ER-TEST: Evaluating Explanation Regularization Methods for NLP Models

Brihi Joshi[♣*] **Aaron Chan**[♣*] **Ziyi Liu**[♣]
Shaoliang Nie[♢] **Maziar Sanjabi**[♢] **Hamed Firooz**[♢] **Xiang Ren**[♣]
[♣]University of Southern California [♢]Meta AI
{brihijos, chanaaro, zliu2803, xiangren}@usc.edu
{snie, maziars, mhfirooz}@fb.com

Abstract

Neural language models' (NLMs') reasoning processes are notoriously hard to explain. Recently, there has been much progress in automatically generating machine rationales of NLM behavior, but less in utilizing the rationales to improve NLM behavior. For the latter, explanation regularization (ER) aims to improve NLM generalization by pushing the machine rationales to align with human rationales. Whereas prior works primarily evaluate such ER models via in-distribution (ID) generalization, ER's impact on out-of-distribution (OOD) is largely underexplored. Plus, little is understood about how ER model performance is affected by the choice of ER criteria or by the number/choice of training instances with human rationales. In light of this, we propose ER-TEST, a protocol for evaluating ER models' OOD generalization along three dimensions: (1) unseen datasets, (2) contrast set tests, and (3) functional tests. Using ER-TEST, we study three key questions: (A) Which ER criteria are most effective for the given OOD setting? (B) How is ER affected by the number/choice of training instances with human rationales? (C) Is ER effective with distantly supervised human rationales? ER-TEST enables comprehensive analysis of these questions by considering a diverse range of tasks and datasets. Through ER-TEST, we show that ER has little impact on ID performance, but can yield large gains on OOD performance w.r.t. (1)-(3). Also, we find that the best ER criterion is task-dependent, while ER can improve OOD performance even with limited and distantly-supervised human rationales.

1 Introduction

Neural language models (NLMs) have achieved state-of-the-art performance on a broad array of natural language processing (NLP) tasks (Devlin et al., 2018; Liu et al., 2019). Even so, NLMs' reasoning processes are notoriously opaque (Rudin, 2019; Doshi-Velez and Kim, 2017; Lipton, 2018), which has spurred significant interest in designing algorithms to automatically explain NLM behavior (Denil et al., 2014; Sundararajan et al., 2017; Camburu et al., 2018; Rajani et al., 2019; Luo et al., 2021). The majority of this work has focused on *rationale extraction*, which explains a NLM's output on a given task instance by highlighting the input tokens that most influenced the output (Denil et al., 2014; Sundararajan et al., 2017; Li et al., 2016; Jin et al., 2019; Lundberg and Lee, 2017).

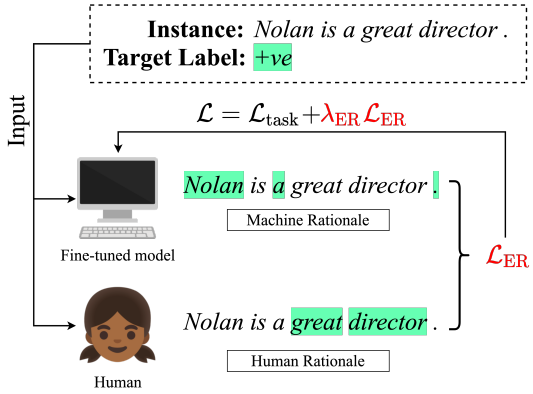

Figure 1: **Explanation Regularization:** Given an instance and a target label, we can use rationale extractors (See Section 2) to generate machine rationales from a model $\mathcal{F}$. Furthermore, human rationales are collected from annotators. Explanation Regularization (ER) aligns machine rationales to human rationales with a loss term, $\mathcal{L}_{\text{ER}}$, which is then used to refine $\mathcal{F}$.

Recently, a number of works have investigated how *machine rationales* produced by rationale extraction algorithms can be operationalized to improve NLM decision-making (Hase and Bansal, 2021) (See Figure 1). Almost all prior works are based on *explanation regularization* (ER), which aims to improve NLM generalization by regularizing the NLM to yield machine rationales that align with *human rationales* (Ross et al., 2017; Huang et al., 2021; Ghaeini et al., 2019; Zaidan and Eis-

[*]Equal contribution.

ner, 2008; Kennedy et al., 2020; Rieger et al., 2020; Liu and Avci, 2019). Human rationales can be created by annotating each training instance individually (Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019) or by applying task-level human priors across all training instances (Kennedy et al., 2020; Rieger et al., 2020; Ross et al., 2017; Liu and Avci, 2019).

Although prior works primarily evaluate such ER models via in-distribution (ID) generalization (Zaidan and Eisner, 2008; Lin et al., 2020; Huang et al., 2021), out-of-distribution (OOD) generalization is more crucial in many real-world settings (Chrysostomou and Aletras, 2022; Ruder, 2021), yet ER's impact on OOD generalization is largely underexplored (Ross et al., 2017; Kennedy et al., 2020). Plus, despite them being major factors in ER, little is understood about how ER model performance is affected by the choice of ER criterion or by the number/choice of training instances with human rationale supervision. In light of this, we propose **ER-TEST**, a protocol for evaluating ER models' OOD generalization along three dimensions: (1) *unseen datasets*, (2) *contrast set tests*, and (3) *functional tests*. For (1), ER-TEST assesses ER models' task performance on datasets beyond their training distribution. For (2), ER-TEST assesses ER models' sensitivity to counterfactual instances created by perturbing existing datasets. For (3), ER-TEST assesses ER models' basic linguistic capabilities (*e.g.,* perception of word/phrase sentiment, robustness to typos) expected for the given task.

Using ER-TEST, we study three key questions: (A) Which *ER criterion* are most effective for the given OOD setting? (B) How is ER affected by the *number/choice of training instances* with human rationales? (C) Is ER effective with *distantly supervised* human rationales? ER-TEST enables comprehensive analysis of these questions by considering a diverse range of text classification tasks and datasets. Through ER-TEST, we show that ER has little impact on ID performance (Sec. 5.3.1, 5.4.1), but can yield large gains on OOD performance (Sec. 5.3.2, 5.4.2) w.r.t. (1)-(3). Also, we find that the best ER criterion is task-dependent (Sec. 5.3), while ER can improve OOD performance even with limited human rationale supervision (Sec. 5.4). Furthermore, we show that ER can improve OOD performance using distantly supervised human rationales (Sec. 5.5).

2 Background

Text Classification Let $\mathcal{F}$ be a NLM task model for M-class text classification. In modern NLP systems, $\mathcal{F}$ usually has a BERT-style architecture (Devlin et al., 2018), consisting of a Transformer encoder (Vaswani et al., 2017) followed by a linear layer with softmax classifier. Let $\mathbf{x}_i = [x_i^t]_{t=1}^n$ be the n-token input sequence (*e.g.,* a sentence) for task instance i. For sequence classification, $\mathcal{F}$ predicts a class for sequence $\mathbf{x}_i$, so let $\mathcal{F}(\mathbf{x}_i) \in \mathbb{R}^M$ be the logits for $\mathbf{x}_i$. Let $y_i = \arg\max_c \mathcal{F}(\mathbf{x}_i)_c$ denote $\mathcal{F}$'s predicted class for $\mathbf{x}_i$. For token classification, $\mathcal{F}$ predicts a class for each token x_i^t, so let $\mathcal{F}(\mathbf{x}_i) \in \mathbb{R}^{n \times M}$ be the logits for the n tokens in $\mathbf{x}_i$. Let $y_{i,t} = \arg\max_c \mathcal{F}(\mathbf{x}_i)_{t,c}$ denote $\mathcal{F}$'s predicted class for x_i^t. Let $y_i = [y_{i,t}]_{t=1}^n$ collectively denote all of $\mathcal{F}$'s predicted token classes for $\mathbf{x}_i$.

Rationale Extraction Given $\mathcal{F}$, $\mathbf{x}_i$, and y_i, the goal of rationale extraction is to output machine rationale $\mathbf{r}_i = [r_i^t]_{t=1}^n$, such that each $r_i^t \in [0, 1]$ is an *importance score* indicating how strongly token x_i^t influenced $\mathcal{F}$ to predict class y_i. Let $\mathcal{G}$ denote a rationale extractor, such that $\mathbf{r}_i = \mathcal{G}(\mathcal{F}, \mathbf{x}_i, y_i)$. $\mathcal{G}$ first computes raw importance scores $\mathbf{s}_i \in \mathbb{R}^n$, then normalizes $\mathbf{s}_i$ as probabilities $\mathbf{r}_i$ using the sigmoid function. In general, $\mathcal{G}$ can be a heuristic or learned function, but we focus on heuristic $\mathcal{G}$ in this work, since they are more common (Luo et al., 2021). Broadly, heuristic $\mathcal{G}$'s can either be a gradient-based, that assign importance scores based on gradient changes in $\mathcal{F}$ (Sundararajan et al., 2017; Sanyal and Ren, 2021; Shrikumar et al., 2017), sampling-based, that assign important scores based on the neighbours/context of a given token (Zeiler and Fergus, 2013; Jin et al., 2019), or attention-based, that use the attention-scores or a function of them to assign importance scores. (Ding and Koehn, 2021)

Explanation Regularization (ER) However, $\mathcal{G}$ can also be used to compute machine rationales w.r.t. other classes besides y_i, *e.g.,* target class $\dot{y}_i$. Let $\hat{\mathbf{r}}_i$ denote the machine rationale for $\mathbf{x}_i$ w.r.t. $\dot{y}_i$. Given $\hat{\mathbf{r}}_i$ obtained via $\mathcal{G}$ and $\mathcal{F}$, many works have explored ER, in which $\mathcal{F}$ is regularized such that $\hat{\mathbf{r}}_i$ aligns with human rationale $\dot{\mathbf{r}}_i$ (Zaidan and Eisner, 2008; Lin et al., 2020; Rieger et al., 2020; Ross et al., 2017). Typically, $\dot{\mathbf{r}}_i$ is a binary vector, where ones and zeros indicate positive (important) and negative (unimportant) tokens, respectively. ER's inductive bias pushes $\mathcal{F}$ to solve the task in a way

that follows the human reasoning process given by $\dot{\mathbf{r}}_i$, which ideally provides denser learning signal for improving $\mathcal{F}$'s generalization.

We formalize the ER loss as: $\mathcal{L}_{\text{ER}} = \Phi(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i)$, where Φ is an ER criterion measuring the alignment between $\hat{\mathbf{r}}_i$ and $\dot{\mathbf{r}}_i$. Thus, the full learning objective is: $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{ER}}\mathcal{L}_{\text{ER}}$, where $\mathcal{L}_{\text{task}}$ is the task loss (*e.g.,* cross-entropy loss) $\lambda_{\text{ER}} \in \mathbb{R}$ is the *ER strength* (*i.e.,* loss weight) for $\mathcal{L}_{\text{ER}}$. Let $\gamma_{\text{ER}} > 0$ be the *rationale scaling factor*, used to scale $\hat{\mathbf{s}}_i$ prior to sigmoid normalization. If the magnitudes of the $\hat{\mathbf{s}}_i$ scores are lower, then the $\hat{\mathbf{r}}_i$ scores will be closer to 0.5 (*i.e.,* lower confidence). However, scaling $\hat{\mathbf{s}}_i$ by $\gamma_{\text{ER}} > 1$ will increase the magnitude of $\hat{\mathbf{s}}_i$, yielding $\hat{\mathbf{r}}_i$ scores closer to 0 or 1 (*i.e.,* higher confidence). Though there are many possible choices for Φ, it is presently unclear how different Φ impact training and when certain Φ should be preferred. This limits our ability to use ER in real-world settings.

3 ER-TEST

Existing works primarily evaluate ER models via ID generalization (Zaidan and Eisner, 2008; Lin et al., 2020; Huang et al., 2021), though a small number of works have done auxiliary evaluations of OOD generalization (Ross et al., 2017; Kennedy et al., 2020; Rieger et al., 2020). However, these OOD evaluations have been relatively small-scale, only covering a narrow range of OOD generalization aspects, ER criteria, training settings, tasks, and datasets. As a result, little is understood about ER's impact on OOD generalization. To address this gap, we propose ER-TEST, a unified benchmark for evaluating ER models' OOD generalization along three dimensions: (1) unseen datasets; (2) contrast set tests; and (3) functional tests.

3.1 ID Generalization

While ER-TEST's main focus is on evaluating OOD generalization, ER-TEST also considers ID generalization as a baseline evaluation. Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}_{i=1}^{N}$ be a M-class text classification dataset, where $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ are the text inputs, $\mathcal{Y} = \{\dot{y}_i\}_{i=1}^{N}$ are the target classes, and N is the number of instances $(\mathbf{x}_i, \dot{y}_i)$ in $\mathcal{D}$. We call $\mathcal{D}$ the ID dataset. Assume $\mathcal{D}$ can be partitioned into train set $\mathcal{D}_{\text{train}}$, dev set $\mathcal{D}_{\text{dev}}$, and test set $\mathcal{D}_{\text{test}}$, where $\mathcal{D}_{\text{test}}$ is an ID test set for $\mathcal{D}$. After using ER to train $\mathcal{F}$ on $\mathcal{D}_{\text{train}}$, we measure $\mathcal{F}$'s task performance on the ID test set $\mathcal{D}_{\text{test}}$. Note that this is a standard protocol used by existing works to evaluate ER models (Zaidan and Eisner, 2008; Rieger et al., 2020; Liu and Avci, 2019; Ross et al., 2017; Huang et al., 2021; Ghaeini et al., 2019; Kennedy et al., 2020).

3.2 OOD Generalization

To assess $\mathcal{F}$'s generalization ability when using ER, we consider various OOD settings. Given $\mathcal{D}$, let $\tilde{\mathcal{D}}_{\text{test}}$ denote an OOD test set, with a different distribution from $\mathcal{D}$. While $\mathcal{F}$ is expected to perform well on $\mathcal{D}_{\text{test}}$ (ID), $\mathcal{F}$ should also perform well on $\tilde{\mathcal{D}}_{\text{test}}$ (OOD). For each dimension of OOD generalization, we obtain $\tilde{\mathcal{D}}_{\text{test}}$ in a different manner.

3.2.1 Unseen Datasets

First, we evaluate OOD generalization w.r.t. unseen datasets. Besides $\mathcal{D}$, suppose we also have a set of datasets $\{\tilde{\mathcal{D}}^{(1)}, \tilde{\mathcal{D}}^{(2)}, ...\}$ of the same task as $\mathcal{D}$. Each of these datasets $\tilde{\mathcal{D}}^{(i)}$ has its own train/dev/test sets and a distribution shift from $\mathcal{D}$. After using ER to train $\mathcal{F}$ on $\mathcal{D}_{\text{train}}$, we measure $\mathcal{F}$'s task performance on each OOD test set $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$. In other words, $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$ is obtained by simply taking the test set of existing OOD dataset $\tilde{\mathcal{D}}^{(i)}$. This evaluation is designed to assess whether ER helps $\mathcal{F}$ learn general (*i.e.,* task-level) knowledge representations that can (zero-shot) transfer across datasets.

3.2.2 Contrast Set Tests

Second, we evaluate OOD generalization under meaningful dataset perturbations. Annotation artifacts (Gururangan et al., 2018) are gaps present in a dataset that can lead to misleading interpretations of a model's performance on that dataset. To mitigate this, we evaluate $\mathcal{F}$ on contrast sets (Gardner et al., 2020), which are (mostly) label-changing small perturbations on instances to understand the true local boundary of the dataset. Essentially, they help us understand if $\mathcal{F}$ has learnt any dataset-specific shortcuts.

Given $\tilde{\mathcal{D}}^{(i)}_{\text{test}}(j)$ (a j^{th} instance belonging to an OOD test set $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$), a perturbation function $\beta^{(i)}_p$ is applied to that instance, where p denotes the kind of perturbation taking effect, and it often changes the target label for that instance. For example, p can signify semantic (*e.g.,* changing *tall* to *short*), numeral (*e.g.,* changing *one dog* to *three dogs*), or entities (*e.g.,* changing *dogs* to *cats*). Each perturbation type is specific to the dataset it is being created for, so that instance labels are changed in a meaningful manner. The resulting set of instances $\mathcal{C}^{\langle\rangle} = \beta^{(i)}_p(\tilde{\mathcal{D}}^{(i)}_{\text{test}}(j))\forall j, p$ are termed as a *contrast*

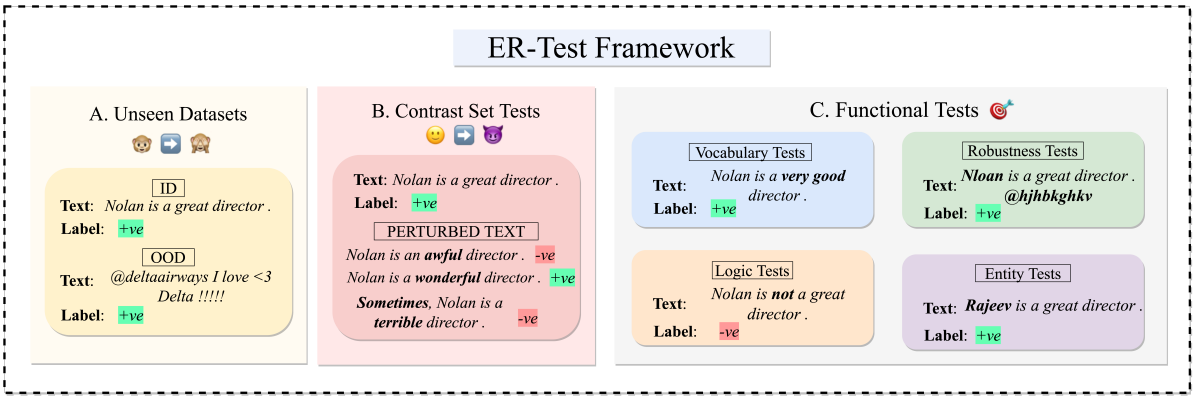

Figure 2: **ER-TEST Framework** - Apart from existing ID evaluations of ER criteria, ER-TEST evaluates ER's impact on OOD generalization along three dimensions: A. Unseen datasets, B. Contrast set tests and C. Functional tests. Examples of individual functional tests shown here are not exhaustive. See Section 3 and Appendix A.4 for more details.

set for that dataset. Based on the way they are created, contrast sets are a property of the dataset, and are not created to explicitly challenge $\mathcal{F}$ (unlike adversarial examples (Gao and Oates, 2019)).

3.2.3 Functional Tests

Third, we evaluate OOD generalization w.r.t. functional tests (Ribeiro et al., 2020; Li et al., 2020). Unlike contrast sets which are designed to test artifacts present in a dataset, functional tests are used to provide *'zoomed-in'* insights about specific linguistic capabilities (like changes in the vocabulary, adding negations to instances, etc). Furthermore, contrast sets are created by perturbing a reference real-world dataset, whereas, functional tests evaluate specific capabilities with the help of template-generated synthetic instances.

If ER consistently improves $\mathcal{F}$'s performance on such tests, then we can have higher confidence that ER is a useful inductive bias for OOD generalization for that given capability Across all tasks, ER-TEST considers four categories of stress tests, which are adopted from CheckList (Ribeiro et al., 2020). Each test is described below.

Vocabulary Tests Vocabulary tests are used to evaluate $\mathcal{F}$'s capability to address changes in the vocabulary of the text, and is particularly diverse w.r.t the parts-of-speech it caters to. For example, certain vocabulary tests evaluate the relationship (taxonomy) between different nouns in a sentence, whereas some swap the modifiers or the verbs present in a sentence in a meaningful manner based on the task at hand, to capture $\mathcal{F}$'s targeted performance towards such changes (Ribeiro et al., 2020).

Robustness Tests Robustness tests evaluate $\mathcal{F}$'s behavior under character-level edits to words in a sentence, keeping the rest of the context same so as to not change the overall prediction. They include testing against typos as well as contractions in words, as well as addition of tokens that are irrelevant for the downstream task (like URLs or gibberish like Twitter handles). (Jones et al., 2020; Wang et al., 2020)

Logic Tests Testing $\mathcal{F}$'s reasoning capabilities towards logical changes in a sentence is also important to evaluate its reliance on shortcut-patterns. These tests perturb sentences in a logical manner (by adding or removing negations, or purposefully inducing contradictions) that also change the target label in the same manner. (Talman and Chatzikyriakidis, 2018; McCoy et al., 2019)

Entity Tests For certain tasks, named entities like numbers, locations and proper nouns are not relevant for predicted a target label, and are often a source of gender or demographic biases (Mishra et al.; Mehrabi et al., 2020). Entity tests measure $\mathcal{F}$'s sensitivity towards changed in named entities such that the overall context as well as the task label remains the same (Ribeiro et al., 2020).

3.3 Tasks and Datasets

To evaluate ER models, ER-TEST considers a diverse set of sequence and token classification tasks. For each, task ER-TEST provides one ID dataset (annotated with human rationales) and multiple OOD datasets. Compared to prior works, ER-TEST's task/dataset diversity enables more extensive analysis of ER model generalization.

First, we have sentiment analysis, using SST

(movie reviews) (Socher et al., 2013; Carton et al., 2020) as the ID dataset. For OOD datasets, we use Yelp (restaurant reviews) (Zhang et al., 2015), Amazon (product reviews) (McAuley and Leskovec, 2013), and Movies (movie reviews) (Zaidan and Eisner, 2008; DeYoung et al., 2019). Movies' inputs are much longer than the other three datasets'. For contrast set tests, we use an OOD contrast set for sentiment analysis released by the authors of the original paper (Gardner et al., 2020), which are created for the Movies dataset. Furthermore, for functional tests, we use an OOD test suite (flight reviews) from the `CheckList` (Ribeiro et al., 2020) which contains both template-based instances to test linguistic capabilities, as well as real-world data (tweets).

Second, we have natural language inference (NLI), using e-SNLI (Camburu et al., 2018; DeYoung et al., 2019) as the ID dataset. For the OOD dataset, we use MNLI (Williams et al., 2017). e-SNLI contains only image captions, while MNLI contains both written and spoken text, covering various topics, styles, and formality levels. For NLI, we also use an OOD contrast set created for the MNLI dataset (Li et al., 2020). Functional tests for NLI are generated from the `AllenNLP` test suite (Gardner et al., 2017) for textual entailment.

Third, we have named entity recognition (NER), using CoNLL-2003 (Sang and De Meulder, 2003; Lin et al., 2020) as the ID dataset. For the OOD dataset, we use OntoNotes v5.0 (Pradhan et al., 2013). CoNLL-2003 contains only Reuters news stories, while OntoNotes v5.0 contains text from newswires, magazines, telephone conversations, websites, and other sources.

3.4 Distantly Supervised Human Rationales

Typically, human rationales are created by annotating each training instance individually (Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019). For each training instance, humans are asked to mark tokens that support the gold label as positive, while the remaining tokens are marked as negative. Here, each human rationale is specifically conditioned on the input and gold label for the given instance. However, such *instance-level* human rationales are very expensive to obtain, given the high manual effort per instance.

Alternatively, some works have constructed distantly supervised human rationales by applying *task-level* human priors across all training instances (Kennedy et al., 2020; Rieger et al., 2020; Ross et al., 2017; Liu and Avci, 2019). For example, Kennedy et al. (2020) used a "blacklist" lexicon to distantly supervise human rationales for the hate speech detection task. In the past, hate speech detection models were largely oversensitive to certain group identifier words (*e.g.,* "black", "Muslim", "gay"), almost always predicting hate speech for text containing these words. To address this, they first manually annotated a lexicon of group identifiers that should be ignored for hate speech detection. Then, for all training instances, they automatically marked only tokens belonging to the lexicon as negative (and the rest as positive). By using these human rationales for ER, they trained the NLM to be less biased w.r.t. these group identifiers. For the purpose of our study, we use the lexicons as used by (Jin et al., 2021) to generate distantly-supervised rationales for the Stormfromt (Stf) dataset (de Gibert et al., 2018). Each instance in the Stf dataset is matched to one or more lexicons by simple character-level matching, and the rationales are generated as described above. We evaluate ER methods on OOD hate speech detection datasets like HatEval (Barbieri et al., 2020) and Gab Hate Corpus (GHC) (Kennedy et al., 2018). All of the datasets contain binary labels for hateful and non-hateful content. The Stf dataset is collected from a white-supremacist forum, whereas HatEval instances are tweets and GHC instances are taken from the Gab forum.

4 Analysis Setup

After introducing the ER-TEST framework, we conduct a systematic study of ER through three primary research questions (described below) using ER-TEST. First, we aim to study *which* ER criteria are effective for a given task at hand (Section 4.1). Second, we study ER from a resource constraint perspective, where only a handful of instances can have human rationales annotated. What is key here is to define *how* to select these instances for rationale annotation (Section 4.2). Lastly, we look at human rationales that are distantly-supervised and how ER can help improve performance when dense, instance-level human rationale annotations are not available (Section 4.3).

4.1 *RQ1: Which ER criteria are most effective?*

Compared to existing works, ER-TEST uses a wider range of ER criteria to evaluate ER model

generalization. This provides a more comprehensive picture of ER's impact on both ID and OOD generalization. Also, this can help us understand why certain criteria work well and under what settings they work best. To demonstrate the utility of ER-TEST, we consider five representative ER criteria (*i.e.,* choices of Φ).

Mean Squared Error (MSE) MSE is used in Liu and Avci (2019), Kennedy et al. (2020), and Ross et al. (2017).

$$\Phi_{\text{MSE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \|\hat{\mathbf{r}}_i - \dot{\mathbf{r}}_i\|_2^2 \tag{1}$$

Mean Absolute Error (MAE) MAE is used in Rieger et al. (2020).

$$\Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = |\hat{\mathbf{r}}_i - \dot{\mathbf{r}}_i| \tag{2}$$

Binary Cross Entropy (BCE) BCE loss is used in Chan et al. (2021).

$$\Phi_{\text{BCE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = -\sum_{t=1}^{n} \dot{r}_i^t \log(\hat{r}_i^t) \tag{3}$$

Huber Loss Huber loss (Huber, 1992) is a hybrid of MSE and MAE.

$$\Phi_{\text{Huber}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \begin{cases} \frac{1}{2}\Phi_{\text{MSE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i), & \Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) < \delta \\ \delta(\Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{4}$$

Order Loss Recall that the human rationale $\dot{\mathbf{r}}_i$ labels each token as positive (one) or negative (zero). Whereas other criteria generally push positive/negative tokens' importance scores to be as high/low as possible, order loss (Huang et al., 2021) relaxes MSE to merely enforce that all positive tokens' importance scores are higher than all negative tokens' importance scores. This is especially useful if $\dot{\mathbf{r}}_i$ is somewhat noisy, *e.g.,* some positively-labeled tokens should not really be positive.

$$\Phi_{\text{Order}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \sum_{\dot{r}_i^t=1} \left(\min\left(\frac{\hat{r}_i^t}{\max_{\dot{r}_j^t=0} \hat{r}_j^t} - 1, 0 \right) \right)^2 \tag{5}$$

4.2 *RQ2: How is ER affected by the number/choice of train instances with human rationales?*

In real-world applications, it is infeasible to obtain human rationales $\dot{\mathbf{r}}_i$ for all of the instances in the training set, as $\dot{\mathbf{r}}_i$ requires dense annotation (Chiang and Lee, 2022; Kaushik et al., 2019).

Let $\mathcal{S}$ be a subset of train instances for which we have human rationale annotations, $\dot{\mathbf{r}}_i^{\mathcal{S}}$. Therefore, the ER loss $\mathcal{L}_{\text{ER}}^{\mathcal{S}} = \Phi(\hat{\mathbf{r}}_i^{\mathcal{S}}, \dot{\mathbf{r}}_i^{\mathcal{S}})$ and the full learning objective $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{ER}}^{\mathcal{S}} \mathcal{L}_{\text{ER}}^{\mathcal{S}}$, where $\mathcal{L}_{\text{task}}$ is computed on the full dataset as it would normally.

In designing such a system, one needs to carefully select $\mathcal{S}$ that leads to highest performance gains, meanwhile maintaining resource constraints. To select relevant samples to annotate, existing methods use to active-learning based approaches (Schröder and Niekler, 2020). We use ER-TEST to compare three such approaches approaches to select $\mathcal{S}$:

Random Sampling Given a k, we uniformly select $k\%$ of samples from $\mathcal{D}$ to construct $\mathcal{S}$.

Lower Confidence (LC) Sampling Given a k, we select top $k\%$ of samples ordered on the basis of the Lower Confidence criterion. (Zheng and Padmanabhan, 2002)

$$\max_i 1 - \mathcal{P}_\theta(\dot{y}_i | x_i) \tag{6}$$

where $(x_i, \dot{y}_i) \in \mathcal{D}_{\text{train}}$, and θ are the parameters of a model trained on $\mathcal{D}_{\text{train}}$ without ER. In other words, these are the top $k\%$ examples that a model trained without ER is the *least* confident on.

Higher Confidence (HC) Sampling Given a k, we sample the top $k\%$ of samples ordered in the reverse order of lower confidence prioritisation as described above. In other words, these are the top $k\%$ examples that a model trained without ER is the *most* confident on.

4.3 *RQ3: Is ER effective with distantly supervised human rationales?*

In Section 4.2, we enforce constraints in the number of instances to annotate with human rationales, and use ER-TEST to compare different strategies to select such instances. However, annotating each token within each instance is also an expensive task, as described in Section 3.4. One way to overcome this issue is generate human rationales with distant supervision.

Let $\mathcal{L}_{\mathcal{D}}$ be a list of lexicons curated by human annotators, specific to a given dataset $\mathcal{D}$. Let $l(\cdot)$ be an indicator function that searches for a given lexicon list in all the tokens of an instance, and returns a binary representation of the same size as the instance with 1s in places with lexicon matches (0 otherwise). Therefore, we can obtain distantly-supervised human rationales $\dot{\mathbf{r}}_i = \mathbb{1} - l(\mathcal{L}_{\mathcal{D}}, x_i)$.

ER Criteria	Sentiment Analysis				NLI		NER	
	In-Distribution	Out-of-Distribution			In-Distribution	Out-of-Distribution	In-Distribution	Out-of-Distribution
	SST	Amazon	Yelp	Movies	e-SNLI	MNLI	CoNLL-2003	OntoNotes v5.0
None	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)	76.18 (±1.28)	46.15 (±4.38)	77.24 (±0.20)	20.78 (±0.41)
MSE	94.29 (±0.05)	90.58 (±0.77)	92.17 (±0.64)	90.00 (±5.63)	78.98 (±1.00)	54.23 (±2.67)	78.02 (±0.69)	21.60 (±0.46)
MAE	94.11 (±0.38)	**92.02 (±0.25)**	**94.55 (±0.30)**	**95.50 (±1.32)**	78.77 (±1.01)	52.41 (±4.50)	**78.34 (±0.81)**	**21.73 (±0.31)**
BCE	94.15 (±0.53)	90.70 (±1.19)	91.82 (±2.30)	92.00 (±6.98)	79.07 (±0.83)	53.68 (±4.15)	64.53 (±13.22)	17.32 (±3.59)
Huber	94.19 (±0.19)	90.43 (±1.45)	92.38 (±2.11)	91.83 (±3.75)	78.99 (±0.81)	53.97 (±3.11)	77.83 (±1.09)	21.38 (±0.16)
Order	**94.37 (±0.11)**	89.47 (±2.71)	87.95 (±6.36)	84.50 (±10.15)	**79.11 (±0.87)**	**55.26 (±3.56)**	72.62 (±5.01)	19.14 (±1.75)

Table 1: **ID/OOD Task Performance (Instance-Based Human Rationales).** This table enlists the ID and OOD performance of different ER criteria (MSE, MAE, BCE, Huber, Order) and compares them to a setting without ER (None). All models (with or without ER) are trained on the ID dataset and evaluated on the ID and OOD datasets without the need of machine or human rationales. Metrics displayed here (higher the better) for sentiment analysis is Accuracy and Macro F1 for NLI and NER.

We can then study the effectiveness of ER methods as detailed in **RQ1** (Section 4.1) in this setting.

[**Brihi**: I am unsure what else to add in here...]

5 Experiments

5.1 Implementation Details

For the NLM architecture, we use BigBird-Base (Zaheer et al., 2020), in order to handle input sequences of up to 4096 tokens. For all results, we report the mean over three seeds, as well as the standard deviation. By default, we use a learning rate of $2e{-}5$ and effective batch size of 32. For ER, there are many possible choices of rationale extractor $\mathcal{G}$, but evaluating all of these choices would be prohibitive. Also, evaluating $\mathcal{G}$ is orthogonal to ER-TEST's goal of evaluating ER criteria Φ. Thus, as a proof of concept, we use the Input*Grad algorithm (Denil et al., 2014) as $\mathcal{G}$ in all experiments, given its popularity and computational efficiency (Bastings and Filippova, 2020; Luo et al., 2021). We leave investigation of other $\mathcal{G}$ for future work.

5.2 Intrinsic Evaluation of ER

ER in general is sensitive to certain hyperparameters for yielding meaningful training curves and actually attaining alignment between machine and human rationales. Due to a large set of tunable hyperparameters, running all configurations of ER are not feasible. Therefore, we intrinsically evaluate hyperparameter configurations by assessing the loss curves (which model alignment between machine and human rationales) w.r.t different hyperparameters values. We observe that the acceptable band of learning rates for ER is very narrow, and we use $2e{-}5$ in all of our experiments. Furthermore, we also observe that setting $\lambda_{\text{ER}} = 1$ and $\gamma_{\text{ER}} = 100$ yields the most drop in the loss curves while training, so we use these hyperparameters for the rest of our experiments. We detail these experiments in Appendix A.1.

5.3 *RQ1: Which ER criteria are most effective?*

5.3.1 ID Generalization

In Table 13 (In-Distribution), we display the ID task performance results for sentiment analysis (SST), NLI (e-SNLI), and NER (CoNLL-2003). For SST, we find that all of the ER criteria yield about the same task performance as the None baseline, whereas, all ER criteria also perform similarly for NLI (yielding higher performance than None). For NER, we see more variance in task performance among ER criteria, although the variance is still quite small among the best methods (MSE, MAE, Huber). Here, MAE yields the highest task performance, while BCE yields the lowest by far. Overall, using ID task performance, it is difficult to distinguish between ER criteria and underplays its overall benefits. This motivates us to consider other evaluation metrics.

5.3.2 OOD Generalization

Unseen Datasets In Table 13 (Out-of-Distribution), we display the OOD task performance results for sentiment analysis (Amazon, Yelp, Movies), NLI (MNLI), and NER (OntoNotes v5.0). For sentiment analysis, MAE yields significant gains over all other ER criteria. Meanwhile, despite performing best on SST, Order performs much worse than all other ER criteria here. For NER, MAE still performs best, while MSE and Huber are competitive. Overall, OOD task performance is much better than ID at distinguishing between ER criteria, especially showing ER's improvement over None.

Contrast Set Tests In Table 2 (Contrast Set Analysis), we observe the drop in performance (denoted by Δ) for sentiment analysis (Movies) and NLI

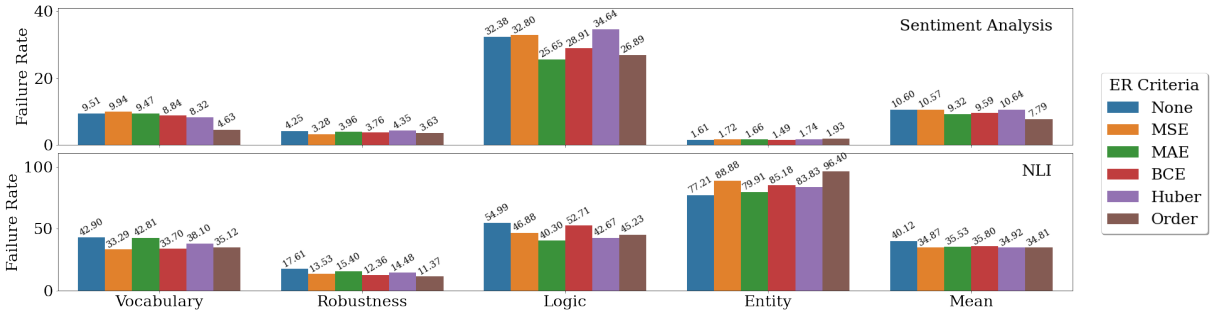

Figure 3: **Functional Tests' Failure Rates (lower the better):** We plot the failure rates of the four functional tests (vocab., robust., logic, entity) as described in Section 3.2.3, as well as the overall failure rate on all of the tests combined (mean). Each of the values are out of 100, but plotted accordingly for visible comparison.

ER Criteria	Contrast Set					
	Sentiment Analysis			NLI		
	Original	Contrast	Δ	Original	Contrast	Δ
None	88.39 (±2.05)	85.11 (±2.72)	-3.28	46.15 (±4.38)	43.73 (±2.81)	-2.42
MSE	88.11 (±2.33)	86.07 (±2.48)	-2.04	54.23 (±2.67)	51.95 (±1.21)	-2.28
MAE	91.12 (±0.59)	89.82 (±1.20)	**-1.30**	52.41 (±4.50)	52.02 (±1.49)	**-0.39**
BCE	89.55 (±1.42)	87.30 (±4.03)	-2.25	53.68 (±4.15)	52.37 (±1.42)	-1.31
Huber	89.20 (±1.67)	86.13 (±1.74)	-3.15	53.97 (±3.11)	52.32 (±1.04)	-1.65
Order	86.00 (±5.27)	83.40 (±6.16)	-2.60	55.26 (±3.56)	52.78 (±0.74)	-2.48

Table 2: **Contrast Set Tests**: Each ER criteria ($\mathcal{F}$) are trained on their ID datasets from Table 13 and evaluated on the OOD original and contrast sets. Δ is the difference in performance of $\mathcal{F}$ between the contrast and original set, and lower the value, better the generalization power of $\mathcal{F}$. A value farther from 0 suggests that $\mathcal{F}$ has learnt shortcuts specific a dataset, which are not generalizable to task that the dataset captures as a whole.

(MNLI) when using a contrast set designed for the given dataset. We observe that for both of the tasks, MAE leads to the least drop in performance.

All of the methods apart from Order yield lower drops than None. All of them also have a higher performance on the original and contrast sets. For sentiment analysis, we observe that Order has the highest variance, and for NLI, it has the highest drop in performance. We believe that some of it can be attributed to the soft-ranking that is imposed by Order, which may be indifferent towards minor label-changing edits, that is observed by the contrast sets.

Functional Tests Figure 3 demonstrates the *failure rates* on functional tests (as listed in Section 3.2.3) of our ID models trained on the sentiment analysis and NLI tasks. We also present the overall aggregated (over each individual tests within the categories mentioned) failure rates.

We observe that apart from the entity-based tests, ER criteria generally have a lower failure rate than None for all of the other tests. For entity-based tests, ER criteria either perform comparably (sentiment analysis) or worse (NLI) than None. Generally, all methods perform well on robustness-based tests, as they have lower failure rates, with order loss having the least. What is important to note is the significant improvement by order loss in vocabulary-based tests than None, even though all of the methods are exposed to the same training set instances. We hypothesize that the biases induced by ER alleviates the shortcuts learnt by None. This is also validated by the overall performance on all of these stress tests, where all of the ER criteria (apart from Huber in sentiment analysis) have lower failure rates than None.

5.4 *RQ2: How is ER affected by the number/choice of train instances with human rationales?*

Table 3 displays the results for our experiments on varying the amount of human rationales available, selected using different sampling methods. For these experiments, we refer to the same training and inference setup we have in Table 13 on sentiment analysis. Furthermore, all hyperparameters are same as that detailed in Section 5.2. We also choose the best performing ER criterion for sentiment analysis from Table 13 (MAE) and base the following results on the same criterion.

5.4.1 ID Generalization

Consistent with the results we observed in Table 13, Table 3 shows us that there are little to no improvements in ID performance with prioritisation methods, however performance for all of them (except LC) is maintained at par with that of None and 100% sampling. This shows that doing ER on selective samples does not degrade ID performance.

5.4.2 OOD Generalization (Unseen Datasets)

Interestingly, we observe distinctions within various prioritisation methods as we look at OOD evaluations in Table 3. In lower-resource scenarios

k (in %)	Selection Method	Sentiment Analysis			
		In-Distribution	Out-of-Distribution		
		SST	Amazon	Yelp	Movies
None	-	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)
100	-	94.11 (±0.38)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)
5	Random	94.36 (±0.05)	91.57 (±0.10)	93.36 (±0.15)	92.39 (±2.50)
	LC	93.14 (±1.97)	90.72 (±0.43)	93.50 (±0.53)	**93.17 (±1.26)**
	HC	94.32 (±0.42)•	91.57 (±0.19)•	93.03 (±0.81)•	91.33 (±3.09)
15	Random	94.46 (±0.21)	90.06 (±1.17)	90.81 (±2.63)	86.22 (±2.94)
	LC	93.48 (±0.80)	90.12 (±2.66)	90.90 (±5.30)	83.67 (±14.02)
	HC	94.39 (±0.27)•	90.38 (±1.12)•	93.48 (±0.64)*	91.33 (±5.11)
50	Random	93.47 (±0.02)	90.28 (±1.42)	91.85 (±2.11)	89.78 (±5.68)
	LC	89.92 (±1.90)	90.75 (±0.78)	93.05 (±0.14)	87.50 (±4.95)
	HC	92.93 (±0.17)□	92.15 (±0.36)*	94.48 (±0.94)*	91.00 (±6.50)*

Table 3: **Instance Prioritisation Methods (with ID/OOD Performance)**: All values are accuracy (higher the better) on sentiment analysis. None corresponds to models trained without ER, where $k = 100\%$ corresponds to no annotation budget. Each of the $k = [5, 15, 50]\%$ have 3 instance prioritisation methods. □ corresponds to cases where HC and Random are significantly similar and greater than LC. $*$ corresponds to cases where HC is significantly greater than Random and greater than LC. • corresponds to cases where all the three methods are significantly similar. (All tests are conducted with $(p < 0.05)$.

(selecting only 5% samples for ER), all of the methods yield similar performance with each other, and outperform None. This implies that doing ER on a smaller subset of instances would instantly yield small improvements over None. As we increase the annotation budget, we observe that model performance declines as we select lower confidence samples, but is maintained or even improves over random while selecting instances with greater confidence. It is important to reiterate here, that samples are prioritized based on the confidence yielded by the None model. This implies that models in general require inductive biases on samples they are *already confident on*, vs. samples they are less confident on to avoid confusion. We plan to study this phenomenon in more detail for other tasks and more constrained annotation budgets as a part of our Future Work (as detailed in Section 6).

5.5 *RQ3: Is ER effective with distantly supervised human rationales?*

5.5.1 ID Generalization

For the task of hate speech detection, we train $\mathcal{F}$ with the Stf dataset. We report all accuracies in Table 4. As it was observed in Section 5.3.1, ER does not lead to a significant improvement in performance for the Stf test set. However, it is important to note that "blacklisting" group identifier lexicons does not lead to a drop in ID performance either. Benefits of "blacklisting" are then observed in OOD generalization.

5.5.2 OOD Generalization

Unseen Datasets We evaluate $\mathcal{F}$ on two OOD datasets, HatEval and GHC. Table 4 shows that while the improvements in HatEval are not significant, there are significant accuracy improvements for the GHC test set, which are due to the Order ER criterion.

Fairness Tests In addition to generic performance metrics like accuracy, we also measure group identifier bias (against the groups detailed by group identifier lexicons) by evaluating the False Positive Rate Difference (FPRD) as shown by (Jin et al., 2021). FPRD is computed as $\sum_z |\text{FPR}_z - \text{FPR}_{\text{overall}}|$, where FPR_z is the false positive rate of all of the test instances mentioning group identifier z, and $\text{FPR}_{\text{overall}}$ is the false positive rate of all the test instance. Essentially, FPRD evaluates if $\mathcal{F}$ is more biased against a given group identifier z, than all of the groups. A lower FPRD value indicates less biased against the listed group identifiers by $\mathcal{F}$.

Table 4 lists the FPRD values of all the ER criteria in ID and OOD datasets. While all other criteria suffer with higher bias then None, we observe that Order criterion consistently leads to the least bias, both in-distribution and out-of-distribution. Furthermore, the reduction in bias is significant when compared to None. Interestingly, Order ER criterion was initially conceived for distantly-supervised rationales (Huang et al., 2021), and the authors of the original paper also demonstrated experiments with rationales generated from lexicons where Order criterion leads to improvements. Our observations are in-line with theirs, and additionally, we also demonstrate its benefit in reducing bias in $\mathcal{F}$.

ER Criteria	Hate Speech Detection					
	In-Distribution		Out-of-Distribution			
	Stf		HatEval		GHC	
	Accuracy ↑	FPRD ↓	Accuracy ↑	FPRD ↓	Accuracy ↑	FPRD ↓
None	89.50 (±0.20)	1.11 (±0.58)	63.68 (±0.78)	1.64 (±0.66)	89.43 (±0.98)	1.09 (±0.12)
MSE	89.46 (±0.21)	2.18 (±0.47)	64.30 (±1.52)	1.99 (±0.26)	88.19 (±0.62)	1.50 (±0.10)
MAE	**89.59 (±0.06)**	1.39 (±0.62)	63.30 (±0.49)	1.80 (±0.59)	88.07 (±1.66)	1.43 (±0.24)
BCE	89.42 (±0.71)	1.87 (±0.45)	63.54 (±0.57)	1.87 (±0.45)	88.99 (±0.83)	1.36 (±0.58)
Huber	89.50 (±0.51)	1.90 (±0.35)	**64.85 (±1.50)**	2.11 (±0.27)	87.77 (±1.21)	1.84 (±0.34)
Order	89.21 (±1.18)	**0.56 (±0.09)**	64.46(±1.18)	**0.92 (±0.92)**	**92.84 (±0.46)**	**0.59 (±0.25)**

Table 4: **ID/OOD Task Performance (Distantly-supervised Human Rationales)**: Higher values for accuracy and lower values for FPRD are considered better. All models displayed are trained on the ID dataset (Stf) with distantly supervised rationales (for ER criteria) and no rationales (for None) and evaluated on ID and OOD test splits.

6 Future Work

Time-based Rationale Annotation Cost Current experiments in selecting instances for ER detailed in Section 5.4 are based on the assumption that each instance takes the same amount of time to annotate. Furthermore, in order to effectively comment about the improvements made by ER under constrained scenarios, there needs to be a comparison between the time taken to annotate explanations (Yao et al., 2021) vs. the time taken to label new training instances, that we aim to evaluate using ER-TEST.

Online ER and connections to human-in-the-loop learning Fine-tuning strategies have shown to distort the underlying data distribution (Kumar et al., 2022), therefore, once $\mathcal{F}$ undergoes ER, its machine rationales differ from before. Currently, ER is being studied in an offline manner – once human rationales are collected, they are used to update model weights. However, what is more effective is to study the effect of ER when applied incrementally, thus improving rationale alignment further.

7 Related Work

ER criteria ER criteria primarily differ in how they obtain human rationale $\dot{\mathbf{r}}_i$ and how they compute machine-human rationale alignment $\Phi(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i)$. First, $\dot{\mathbf{r}}_i$ can be obtained by annotating each training instance individually (Zaidan and Eisner, 2008; Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019; DeYoung et al., 2019) or by applying domain-level human priors across all training instances (Rieger et al., 2020; Ross et al., 2017; Ghaeini et al., 2019; Kennedy et al., 2020; Liu and Avci, 2019). The former approach is more expensive, while the latter approach has more limited applicability since it requires domain knowledge. Second, existing choices of Φ include MSE (Liu and Avci, 2019; Kennedy et al., 2020; Ross et al., 2017), MAE (Rieger et al., 2020), BCE (Chan et al., 2021), order loss (Huang et al., 2021), and KL divergence (Chan et al., 2021). Currently, there is little understanding about how these ER design choices impact OOD generalization, so ER-TEST aims to provide a testbed for conducting such analysis. Beyond ER, Hase and Bansal (2021) presents a more general study about how models can learn from explanations, claiming that explanations are best used as model inputs.

Evaluating ER criteria Existing works have primarily evaluated ER models via ID generalization (Zaidan and Eisner, 2008; Lin et al., 2020; Huang et al., 2021), which only captures one aspect of ER's impact. Meanwhile, a few works have considered auxiliary evaluations — *e.g.,* machine-human rationale alignment (Huang et al., 2021; Ghaeini et al., 2019), task performance on unseen datasets (Ross et al., 2017; Kennedy et al., 2020), social group fairness (Rieger et al., 2020; Liu and Avci, 2019). However, such evaluations are uncommon and relatively small-scale, only covering a narrow range of OOD generalization aspects, ER criteria, tasks, and datasets. These limitations make it difficult to thoroughly compare ER criteria, analyzing why they work and when they work best. To address these limitations, ER-TEST provides a unified benchmark for evaluating multiple aspects of OOD generalization, across a diverse range of ER criteria, tasks, and datasets.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? *arXiv preprint arXiv:2010.05607*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *arXiv preprint arXiv:1812.01193*.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. *arXiv preprint arXiv:2010.04736*.

Aaron Chan, Jiashu Xu, Boyuan Long, Soumya Sanyal, Tanishq Gupta, and Xiang Ren. 2021. Salkg: Learning from knowledge graph explanations for commonsense reasoning. *Advances in Neural Information Processing Systems*, 34.

Cheng-Han Chiang and Hung-yi Lee. 2022. Re-examining human annotations for interpretable nlp.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

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

Hang Gao and Tim Oates. 2019. Universal adversarial perturbation for text classification.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *arXiv preprint arXiv:2004.02709*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Reza Ghaeini, Xiaoli Z Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Saliency learning: Teaching the model where to pay attention. *arXiv preprint arXiv:1902.08649*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Hase and Mohit Bansal. 2021. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*.

Quzhe Huang, Shengqi Zhu, Yansong Feng, and Dongyan Zhao. 2021. Exploring distantly-labeled rationales in neural network models. *arXiv preprint arXiv:2106.01809*.

Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

Xisen Jin, Francesco Barbieri, Brendan Kennedy, Aida Mostafazadeh Davani, Leonardo Neves, and Xiang Ren. 2021. On transferability of bias mitigation effects in language model fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3770–3783, Online. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2752–2765, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Brendan Kennedy, Mohammad Atari, Aida M Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs, Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez, and et al. 2018. Introducing the gab hate corpus: Defining and applying hate-based rhetoric to social media posts at scale.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. *arXiv preprint arXiv:2005.02439*.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. Triggerner: Learning with entity triggers as explanations for named entity recognition. *arXiv preprint arXiv:2004.07493*.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Siwen Luo, Hamish Ivison, Caren Han, and Josiah Poon. 2021. Local interpretations for explainable natural language processing: A survey. *arXiv preprint arXiv:2103.11072*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ninareh Mehrabi, Thamme Gowda, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2020. *Man is to Person as Woman is to Location: Measuring Gender Bias in Named Entity Recognition*, page 231–232. Association for Computing Machinery, New York, NY, USA.

Shubhanshu Mishra, Sijun He, and Luca Belli. [link].

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. 2020. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*.

Sebastian Ruder. 2021. Challenges and Opportunities in NLP Benchmarking. `http://ruder.io/nlp-benchmarking`.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Aarne Talman and Stergios Chatzikyriakidis. 2018. Testing the generalization power of neural network models across nli benchmarks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CATgen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining language models with compositional explanations.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *NeurIPS*.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pages 31–40.

Matthew D Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Zhiqiang Zheng and B. Padmanabhan. 2002. On active learning for data acquisition. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 562–569.

A Appendix

A.1 Intrinsic Evaluation: evaluating ER's sensitivity to hyperparameters

When using ER to train $\mathcal{F}$, it is important to assess whether ER exhibits expected training behavior, orthogonally to task performance. If ER improves task performance, this kind of analysis can help us better understand ER's effectiveness. Conversely, if ER does not improve task performance, such analysis can help us identify the problem.

Motivated by this, ER-TEST's intrinsic evaluation is based on *machine-human rationale alignment*, captured by the ER loss $\mathcal{L}_{\text{ER}} = \Phi(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i)$. When using ER, we should generally expect the ER loss to decrease as $\mathcal{F}$ is trained. In practice, this may not always be the case, even when ER leads to slightly higher task performance (which is likely a mirage caused by lucky random seeds)! That is, by definition, non-decreasing ER loss signals ineffective ER usage, since the machine rationales are not becoming more similar to the human rationales. This can stem from a number of issues: *e.g.,* poor choice of ER criteria Φ, improper ER strength λ_{ER}, improper rationale scaling factor γ_{ER}, noisy human rationale $\dot{\mathbf{r}}_i$, insufficient $\mathcal{F}$ capacity. Thus, we measure machine-human rationale alignment as the first step in diagnosing such issues.

Let *ER loss curve* denote a chart which plots $\mathcal{L}_{\text{ER}}$ *vs.* the number of train epochs. For each combination of ER criteria Φ and some training configuration, we plot ER loss curves for the training set. Each component of our intrinsic evaluation varies a different hyperparameter in the training configuration: (A) ER strength λ_{ER}; (B) rationale scaling factor γ_{ER}; and (C) learning rate α. In contrast, prior works do not explore the relationship between $\mathcal{L}_{\text{ER}}$ and these training variables (Huang et al., 2021; Ghaeini et al., 2019).

For intrinsic evaluation, we use ER strength $\lambda_{\text{ER}} = 1$, rationale scaling factor $\gamma_{\text{ER}} = 1$, and learning rate $\alpha = 2\text{e}{-5}$, unless otherwise specified. As a proof of concept, we focus on SST here, but plan to add other datasets in future work.

A.1.1 ER Strength

Fig. 5 displays the ER loss curves for different ER strengths $\lambda_{\text{ER}} = [0.5, 1, 10, 100, 300]$, on SST using MAE. Among the λ_{ER} values, we see that $\lambda_{\text{ER}} = 1$ yields ER loss curves with the greatest decrease (Table 6), signaling good ER optimization.

	Rationale Scaling Factor			
ER criteria	1	10	100	1000
MSE	0.69	4.60	**18.35**	11.41
MAE	0.04	0.40	**1.29**	1.17
BCE	0.10	0.34	0.90	**1.03**
Huber	0.10	7.75	**16.67**	9.30
Order	7.21	9.38	**47.97**	**1.89**

Table 5: **Relative Decrease in ER Loss.** For various ER rationale scaling factors, we report the percentage decrease in ER train loss (on SST), from max point to min point.

	ER Strength				
ER criteria	0.5	1	10	100	300
MSE	0.91	**1.52**	1.41	1.29	1.35
MAE	1.89	**2.01**	1.72	1.80	1.74
BCE	1.99	**2.17**	1.65	1.65	1.75
Huber	1.85	2.09	2.24	2.27	**2.40**
Order	2.15	2.40	1.60	**2.53**	1.89

Table 6: **Relative Decrease in ER Loss.** For various ER strengths, we report the percentage decrease in ER train loss (on SST), from max point to min point.

A.1.2 Rationale Scaling Factor

Fig. 4 displays the ER loss curves for different rationale scale factors $\gamma_{\text{ER}} = [1, 10, 100, 1000]$, on SST. Among the four γ_{ER} values, we see that $\gamma_{\text{ER}} = 100$ yields ER loss curves with the greatest decrease (Table 5), signaling good ER optimization. Meanwhile, although ER works use $\gamma_{\text{ER}} = 1$ by default, we see that $\gamma_{\text{ER}} = 1$ yields nearly flat ER loss curves for all five Φ choices. This suggests poor ER optimization. Based on these results, we fix $\gamma_{\text{ER}} = 100$ for all experiments (Sec. 5), thus greatly reducing the hyperparameter search space (Sec. A.3).

A.1.3 Learning Rate

Here, we obtain similar conclusions, with $\alpha = 2\text{e}{-5}$ yielding the best ER loss curves (Sec. A.1.4).

A.1.4 Learning Rate

Fig. 6 displays the ER loss curves for different learning rates $\alpha = [2\text{e}{-6}, 2\text{e}{-5}, 2\text{e}{-4}]$. Among the three learning rates, we see that $\alpha = 2\text{e}{-5}$ yields the most steadily decreasing ER loss curves.

A.2 ER performance with different hyperparameters

ER Strength *vs.* Task Performance To measure ER's impact on task performance, we plot $\mathcal{F}$'s task performance as a function of ER strength λ_{ER}. This is conducted for ID test sets.

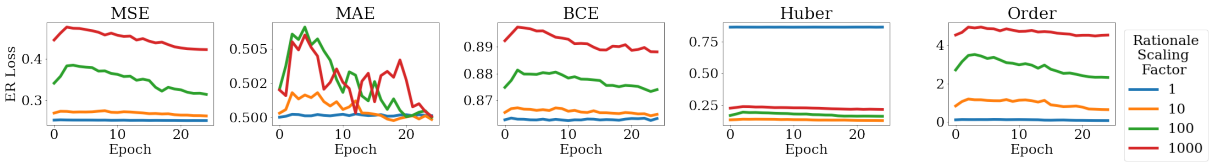

Figure 4: **ER Loss Curves (Rationale Scaling Factor).**

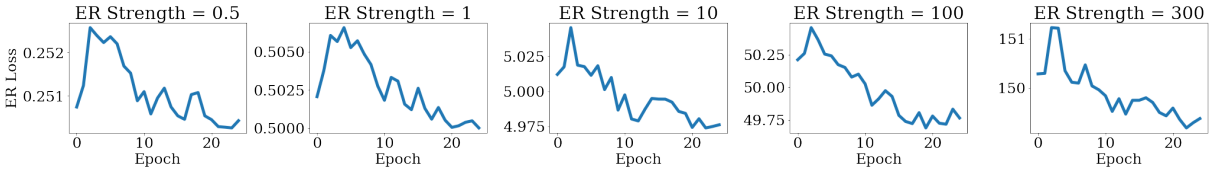

Figure 5: **ER Loss Curves (ER Strength).** Here, we use the MAE criterion.

ER Loss *vs.* Task Performance To measure ER's impact on task performance, we plot $\mathcal{F}$'s task performance as a function of ER loss $\mathcal{L}_{\text{ER}}$. This is conducted for both ID and OOD test sets.

Change in Target Class Confidence Let $\mathcal{F}_{\text{No-ER}}$ and $\mathcal{F}_{\text{ER}}$ denote non-ER-trained (vanilla) and ER-trained NLMs, respectively. For each test instance, we plot $\mathcal{F}_{\text{No-ER}}$'s predicted target class confidence probability *vs.* $\mathcal{F}_{\text{ER}}$'s. Each point in the plot is color-coded by whether ER changes the prediction from correct to incorrect, changes the prediction from incorrect to correct, keeps the prediction as correct, or keeps the prediction as incorrect. The purpose of this plot is to visualize how individual instances' predictions are affected by ER. We conduct this for ID dev sets.

A.2.1 ER Strength *vs.* Task Performance

For each sentiment analysis dataset, Fig. 7 shows task performance for ER strengths $\lambda_{\text{ER}} = [0, 0.5, 1, 10, 100, 300]$, using MAE. Note that $\lambda_{\text{ER}} = 0$ is equivalent to training the NLM without ER (*i.e.,* None in Table 13). For the ID dataset (SST), we see that all ER strengths yield very similar task performance, suggesting that ER has little effect on ID task performance. However, for the OOD datasets (Amazon, Yelp, Movies), task performance generally increases as λ_{ER} increases, showing ER's positive impact on NLM generalization. Overall, based on OOD task performance, we find that $\lambda_{\text{ER}} = [1, 100]$ are the best ER strengths. This aligns with the results of Sec. A.1.1.

A.2.2 ER Loss *vs.* Task Performance

Fig. 8 displays the SST results for ID task performance (accuracy) *vs.* ER loss. For a given ER criterion, each point in the corresponding scatter plot represents the checkpoint at some train epoch of the ER-trained model, evaluated on either the dev set or test set (yielding two point sets). Fitting each point set with linear regression, we find that there is an inverse relationship between task performance and ER loss. In other words, higher machine-human rationale alignment (*i.e.,* low ER loss) corresponds to higher task performance, which validates the usage of ER to improve generalization. Table 7 displays the slopes and R^2 scores of the lines in Fig. 8. The slope indicates the strength of the relationship between machine-human rationale alignment and task performance (lower is better), while the R^2 score indicates how accurately each line fits its corresponding data points. Among the five ER criteria, across dev and test, we find that MAE has the lowest slopes and highest R^2 scores overall, suggesting that using ER with MAE is most effective.

	Dev		Test	
ER criteria	Slope (↓)	R^2 (↑)	Slope (↓)	R^2 (↑)
MSE	-7.48	0.050	-6.75	0.059
MAE	**-128.60**	0.083	**-133.03**	**0.110**
BCE	-17.48	0.003	-56.30	0.040
Huber	-23.59	0.091	-8.40	0.022
Order	-0.49	**0.101**	-0.085	0.004

Table 7: **ER Loss *vs.* Task Performance.** We summarize the line plots in Fig. 8 (ER Loss *vs.* Task Performance), using slope and R^2 score (Sec. A.2.2). Ideally, Fig. 8's lines would have *low slope* and *high* R^2, indicating that ER helps improve task performance. We see that MAE yields the best ER results.

A.2.3 Change in Target Class Confidence

We consider ER with the MAE criterion, trained/evaluated on SST (via dev ID task performance). Fig. 9 visualizes how ER changes each dev instance's target class confidence as a result of ER, color-coding each point w.r.t. how ER changes the model's predicted class for this point. Among in-

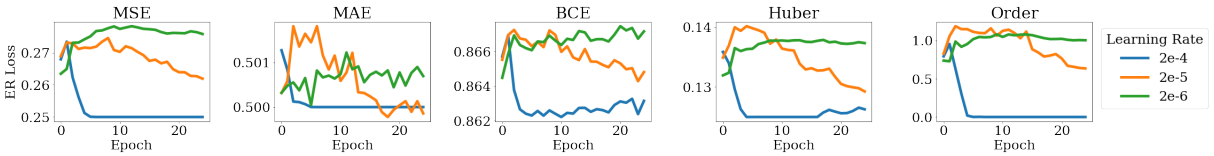

Figure 6: **ER Loss Curves (Learning Rate)**

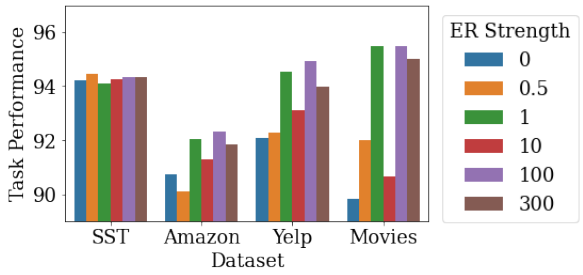

Figure 7: **ER Strength *vs.* Task Performance.** For various combinations of sentiment analysis dataset and ER strength, we plot task performance using MAE.

Percentage of Dev Instances in *incor→cor* Group, Binned by $\mathcal{F}_{\text{No-ER}}$ Target Class Confidence									
0.0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
22.85	26.00	40.38	49.20	28.78	0.00	0.00	0.00	0.00	0.00

Table 8: **Change in Target Class Confidence.** For bins where $\mathcal{F}_{\text{No-ER}}$'s target class confidence is low, there is a higher percentage of instances that are predicted incorrectly/correctly without/with ER. This suggests that instances with low target class confidence are more likely to benefit from ER.

stances for which $\mathcal{F}_{\text{No-ER}}$'s target class confidence is low, there is a higher percentage of instances that are predicted incorrectly/correctly without/with ER (*i.e.,* $incor \rightarrow cor$). This suggests that, for $\mathcal{F}_{\text{No-ER}}$, instances with low target class confidence are more likely to benefit from ER (Table 8). Also, based on the T-test, target class confidence scores are significantly higher ($p < 0.005$) with ER than without.

A.2.4 ER Opportunity Cost

An ER-trained NLM $\mathcal{F}_{\text{task, ER}}$ and a non-ER-trained NLM $\mathcal{F}_{\text{task, No-ER}}$ are likely to yield different outputs given the same inputs. Let $\mathcal{D}^{+}_{\text{ER}} \subseteq \mathcal{D}$ and $\mathcal{D}^{+}_{\text{No-ER}} \subseteq \mathcal{D}$ denote the sets of instances predicted correctly by $\mathcal{F}_{\text{task, ER}}$ and $\mathcal{F}_{\text{task, No-ER}}$, respectively. Ideally, we would have $\mathcal{D}^{+}_{\text{No-ER}} \subset \mathcal{D}^{+}_{\text{ER}}$. This means there is no *opportunity cost* in using ER, as ER increases the number of correct instances without turning any previously-correct incorrect. However, this may not necessarily be the case, so we measure ER's opportunity cost as follows. Let $n^{+}_{\text{ER}} = |\mathcal{D}^{+}_{\text{ER}} \backslash (\mathcal{D}^{+}_{\text{ER}} \cap \mathcal{D}^{+}_{\text{No-ER}})|$ be the number of instances predicted correctly by $\mathcal{F}_{\text{task, ER}}$, but not by $\mathcal{F}_{\text{task, No-ER}}$. Let $n^{+}_{\text{No-ER}} = |\mathcal{D}^{+}_{\text{No-ER}} \backslash (\mathcal{D}^{+}_{\text{No-ER}} \cap \mathcal{D}^{+}_{\text{ER}})|$ be the number of instances predicted correctly by $\mathcal{F}_{\text{task, No-ER}}$, but not by $\mathcal{F}_{\text{task, ER}}$. Then, the opportunity cost of using ER is defined as:

$$o_{\text{ER}} = \frac{n^{+}_{\text{No-ER}} - n^{+}_{\text{ER}}}{|\mathcal{D}|} \tag{7}$$

In practice, instead of defining o_{ER} for all of $\mathcal{D}$, we only consider test sets $\mathcal{D}_{\text{test}}$ and $\tilde{\mathcal{D}}_{\text{test}}$.

ER criteria	Sentiment Analysis			
	In-Domain	Out-of-Domain		
	SST	Amazon	Yelp	Movies
None	0.00 (±0.00)	0.00 (±0.00)	0.00 (±0.00)	0.00 (±0.00)
MSE	0.32 (±1.05)	**-1.25 (±1.20)**	**-2.33 (±4.64)**	-6.50 (±40.66)
MAE	-0.09 (±0.24)	-0.58 (±3.45)	-0.94 (±11.21)	**-7.00 (±40.66)**
BCE	**-0.16 (±0.33)**	0.46 (±4.11)	0.96 (±26.99)	0.16 (±47.72)
Huber	0.12 (±0.42)	0.19 (±2.25)	-1.05 (±4.11)	-4.33 (±37.72)
Order	1.90 (±1.38)	6.98 (±3.87)	19.86 (±45.54)	21.66 (±35.72)

Table 9: **ID/OOD Opportunity Cost.** Lower values are better.

Table 9 displays the opportunity cost results for sentiment analysis. Generally, the opportunity cost results mirror the task performance results in Table 13, such that the methods with highest task performance tend to have the lowest opportunity cost. However, using opportunity cost, the variance is very high for OOD datasets, making it difficult to compare methods. In future work, we plan to modify the opportunity cost metrics to better accommodate OOD settings.

A.3 Efficient hyperparameter tuning with ER-Test

In intrinsic evaluation (Sec. A.1), we used ER loss curves as priors for selecting three key ER hyperparameters (*i.e.,* ER strength λ_{ER}, rationale scaling factor γ_{ER}, learning rate α). In Sec. 5, we assumed a tuning budget that allows only one value for each of λ_{ER}, γ_{ER}, and α. By not tuning these hyperparameters, we greatly reduced our hyperparameter search space. Since ER has little effect on ID task performance, tuning based on ID task performance is unlikely to have helped anyway. ER works better on OOD data, but it also does not make sense to tune based on OOD task performance (otherwise,

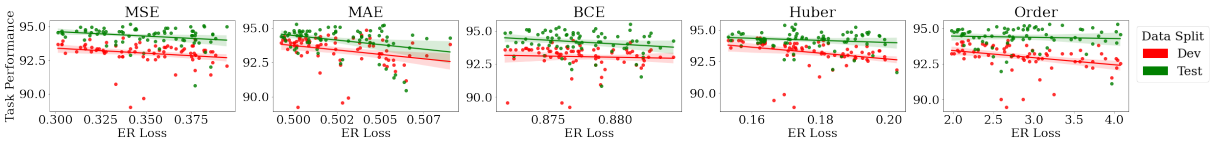

Figure 8: **Task Performance vs. ER Loss**

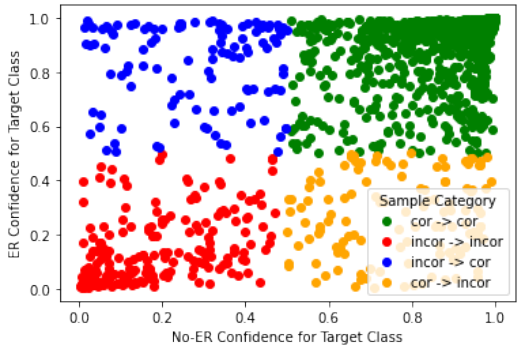

Figure 9: **Change in Target Class Confidence**

ER criteria	Sentiment Analysis (Out-of-Domain)			
	Amazon	Yelp	Movies	Mean
None	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)	XX.XX (±X.XX)
MAE (λ_{ER} = 0.5)	90.12 (±2.98)	92.27 (±3.29)	92.00 (±5.68)	91.46 (±0.91)
MAE (λ_{ER} = 1)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)	94.02 (±2.15)
MAE (λ_{ER} = 10)	91.27 (±0.28)	93.10 (±1.08)	90.67 (±3.79)	91.68 (±1.06)
MAE (λ_{ER} = 100)	**92.33 (±0.28)**	**94.92 (±0.56)**	**95.50 (±0.50)**	**94.25 (±1.89)**
MAE (λ_{ER} = 300)	91.83 (±0.42)	93.97 (±1.28)	95.00 (±0.50)	93.60 (±1.74)
MAE (γ_{ER} = 1)	90.63 (±1.88)	92.32 (±2.23)	88.67 (±4.25)	90.54 (±2.22)
MAE (γ_{ER} = 10)	**92.30 (±1.21)**	93.01 (±2.14)	**96.83 (±1.04)**	**94.07 (±3.89)**
MAE (γ_{ER} = 100)	92.02 (±0.25)	**94.55 (±0.30)**	95.50 (±1.32)	94.02 (±2.15)
MAE (γ_{ER} = 1000)	90.47 (±2.06)	92.80 (±2.90)	92.67 (±6.25)	91.98 (±1.14)
MAE (α = 2e−4)	89.35 (±2.85)	91.23 (±2.84)	93.00 (±2.65)	91.19 (±2.22)
MAE (α = 2e−5)	**92.02 (±0.25)**	**94.55 (±0.30)**	**95.50 (±1.32)**	**94.02 (±2.15)**
MAE (α = 2e−6)	88.60 (±1.60)	83.27 (±6.49)	81.17 (±6.93)	84.34 (±9.70)

Table 10: **Task Performance *vs.* {ER Strength (λ_{ER}), Rationale Scaling Factor (γ_{ER})}.** Higher values are better.

it would not be OOD). Though the ER hyperparameters chosen via intrinsic evaluation generally improved OOD task performance, we seek to verify their effectiveness compared to other possible hyperparameter values.

In Table 10 (in the appendix), we report sentiment analysis OOD (Amazon, Yelp, Movies) task performance, while varying each of the three hyperparameters. We include a Mean column, which averages the Amazon/Yelp/Movies columns. Our hyperparameters chosen via ER loss curves are highlighted in blue. For λ_{ER}, 1 (ours) and 100 yield very similar Mean results, while considerably beating the other three values. For γ_{ER}, we see the same trend for 100 (ours) and 10. For α, 2e−5 (ours) vastly outperforms other values in all columns. These results validate the utility of ER-TEST's intrinsic evaluation for low-resource ER hyperparameter tuning.

A.4 Details for Functional Tests

In this section, we provide details for different functional tests listen in Section 3.2.3. We breakdown each subcategory of functional tests and show performances of different ER criteria on those individual tests. For functional tests on the sentiment analysis task, refer to Table 11. NLI functional tests are listed in Table 12.

A.5 Details for Instance Prioritisation Experiments

In this section, we provide further implementation details for confidence-based instance prioritisation experiments as described in Section 4.2.

Given that we have 3-seed runs for the None model in Table 13, we extract the confidence scores based on the given metric (LC or HC), and then average these confidence scores across the 3 seed runs to obtain a single score for every instance. This process is done for training set instances only. This is followed by ranking each instance by the aggregated confidence metric and selecting the top $k\%$ of samples from this ranking. For experiments with random sampling based prioritisation, we generate 3 random subsets selected in a uniform manner.

While training in this setting, we ensure that within each batch, certain (one third to be specific) set of instances have available rationales. For these instances, we calculate the ER loss $\mathcal{L}_{\text{ER}}$, whereas, for the rest of the instances in the batch, we compute the task loss $\mathcal{L}_{task}$. All prioritisation settings are trained with 3 different model seeds and the aggregated results for ID and OOD datasets are shown in Table 3.

Capability	Test Type	ER criteria					
		None	MSE	MAE	BCE	Huber	Order
Vocabulary	Sentiment-laden words in context	1.20 (±0.74)	0.60 (±0.16)	1.27 (±0.84)	1.00 (±0.86)	1.13 (±0.50)	0.80 (±0.28)
	Change Neutral words with BERT	5.59 (±0.16)	5.13 (±0.90)	5.40 (±0.28)	5.67 (±0.68)	5.67 (±0.74)	5.60 (±1.63)
	Intensifiers	2.13 (±1.63)	1.80 (±0.16)	1.40 (±0.16)	2.67 (±0.77)	2.67 (±0.96)	1.60 (±0.65)
	Reducers	23.85 (±7.18)	35.00 (±46.01)	27.38 (±5.95)	25.00 (±25.00)	17.46 (±13.65)	0.77 (±0.43)
	Add +ve phrases	1.40 (±0.28)	2.33 (±1.84)	0.67 (±0.50)	1.27 (±1.00)	2.33 (±1.76)	2.07 (±1.52)
	Add -ve phrases	22.86 (±7.43)	14.80 (±1.40)	20.67 (±4.07)	17.40 (±3.64)	20.67 (±3.35)	16.93 (±1.91)
Robustness	Adding Random URLs and Handles	9.80 (±0.48)	7.27 (±2.23)	9.07 (±1.80)	7.87 (±2.76)	10.27 (±0.9)	9.6 (±2.47)
	Punctuations	3.93 (±0.89)	1.93 (±0.41)	3.00 (±1.02)	2.87 (±0.19)	3.80 (±0.28)	2.67 (±0.34)
	Typos	2.60 (±0.90)	2.53 (±0.82)	2.60 (±0.57)	3.13 (±0.90)	2.60 (±0.75)	2.00 (±0.86)
	2 Typos	3.93 (±0.65)	3.87 (±1.24)	4.27 (±0.5)	4.13 (±1.2)	4.6 (±0.43)	3.33 (±0.25)
	Contractions	1.00 (±0.00)	0.80 (±0.33)	0.87 (±0.25)	0.80 (±0.43)	0.47 (±0.09)	0.53 (±0.50)
Logic	Negatives	5.20 (±2.75)	4.27 (±1.65)	4.47 (±3.07)	4.47 (±1.75)	3.93 (±1.57)	5.67 (±1.68)
	Non-negatives	59.73 (±9.48)	59.00 (±15.81)	37.47 (±10.41)	63.27 (±17.61)	59.07 (±14.97)	45.87 (±24.13)
	Negation of positive with neutral stuff in the middle	32.2 (±14.65)	35.13 (±1.91)	35.00 (±16.52)	19.00 (±8.66)	40.93 (±4.31)	29.13 (±10.60)
Entity	Change Names	0.70 (±0.14	1.91 (±0.71)	1.11 (±0.51)	0.81 (±0.14)	1.61 (±0.62)	1.91 (±1.51)
	Change Locations	3.33 (±0.74)	2.73 (±1.15)	3.40 (±0.86)	3.07 (±1.79)	3.00 (±0.33)	3.20 (±1.57)
	Change Numbers	0.80 (±0.00)	0.53 (±0.34)	0.47 (±0.41)	0.60 (±0.33)	0.60 (±0.43)	0.67 (±0.81)

Table 11: **Functional Tests:** Sentiment Analysis

Capability	Test Type	ER criteria					
		None	MSE	MAE	BCE	Huber	Order
Vocabulary	Antonym in Hypothesis	71.66 (±20.98)	64.77 (±21.97)	84.55 (±11.53)	65.88 (±21.40)	74.77 (±20.41)	62.55 (±13.16)
	Synonym in Hypothesis	32.61 (±7.41)	24.11 (±7.62)	30.11 (±6.42)	25.88 (±6.86)	30.77 (±7.07)	29.27 (±6.95)
	Supertype in Hypothesis	24.44 (±15.95)	11.00 (±3.62)	13.77 (±6.71)	9.31 (±5.90)	8.77 (±8.06)	13.55 (±7.10)
Robustness	Punctuation	14.55 (±4.13)	9.44 (±2.79)	11.33 (±1.63)	8.11 (±1.19)	10.00 (±2.58)	9.88 (±2.51)
	Typo	15.88 (±3.44)	10.22 (±3.04)	12.33 (±1.63)	9.66 (±2.10)	10.88 (±2.68)	10.77 (±2.52)
	2 Typos	15.33 (±3.68)	9.77 (±1.81)	12.00 (±1.76)	9.44 (±2.31)	11.11 (±2.99)	10.00 (±2.66)
	Contractions	24.69 (±6.98)	24.69 (±8.72)	25.92 (±9.07)	22.22 (±9.07)	25.92 (±7.40)	14.81 (±5.23)
Logic	Negation in the Hypothesis	50.88 (±32.25)	27.77 (±37.24)	9.77 (±15.66)	41.33 (±41.54)	15.22 (±28.77)	18.44 (±23.21)
	Induce Contradiction	99.88 (±0.31)	98.54 (±3.78)	91.69 (±20.37)	98.65 (±2.56)	98.42 (±4.44)	99.88 (±0.31)
	Same Premise and Hypothesis	14.22 (±8.63)	14.33 (±10.14)	19.44 (±12.12)	18.16 (±12.69)	14.38 (±9.23)	17.38 (±10.16)
Entity	Switch one Entity in the Hypothesis	77.21 (±39.57)	88.88 (±24.11)	79.91 (±22.20)	85.18 (±30.04)	83.83 (±24.25)	96.40 (±4.85)

Table 12: **Functional Tests:** NLI

ER Criteria	Sentiment Analysis				NLI		NER		Hate Speech Detection		
	In-Distribution	Out-of-Distribution			In-Distribution	Out-of-Distribution	In-Distribution	Out-of-Distribution	In-Distribution	Out-of-Distribution	
	SST	Amazon	Yelp	Movies	e-SNLI	MNLI	CoNLL-2003	OntoNotes v5.0	Stf.	HatEval	GHC
None	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)	76.18 (±1.28)	46.15 (±4.38)	77.24 (±0.20)	20.78 (±0.41)	89.50 (±0.20)	63.68 (±0.78)	89.43 (±0.98)
MSE	94.29 (±0.05)	90.58 (±0.77)	92.17 (±0.64)	90.00 (±5.63)	78.98 (±1.00)	54.23 (±2.67)	78.02 (±0.69)	21.60 (±0.46)	89.46 (±0.21)	64.30 (±1.52)	89.43 (±0.98)
MAE	94.11 (±0.38)	**92.02 (±0.25)**	**94.55 (±0.30)**	**95.50 (±1.32)**	78.77 (±1.01)	52.41 (±4.50)	**78.34 (±0.81)**	**21.73 (±0.31)**	**89.59 (±0.06)**	63.30 (±0.49)	88.07 (±1.66)
BCE	94.15 (±0.53)	90.70 (±1.19)	91.82 (±2.30)	92.00 (±6.98)	79.07 (±0.83)	53.68 (±4.15)	64.53 (±13.22)	17.32 (±3.59)	89.42 (±0.71)	63.54 (±0.57)	88.99 (±0.83)
Huber	94.19 (±0.19)	90.43 (±1.45)	92.38 (±2.11)	91.83 (±3.75)	78.99 (±0.81)	53.97 (±3.11)	77.83 (±1.09)	21.38 (±0.16)	89.50 (±0.51)	**64.85 (±1.50)**	87.77 (±1.21)
Order	**94.37 (±0.11)**	89.47 (±2.71)	87.95 (±6.36)	84.50 (±10.15)	**79.11 (±0.87)**	**55.26 (±3.56)**	72.62 (±5.01)	19.14 (±1.75)	89.21 (±1.18)	64.46 (±1.18)	**92.84 (±0.46)**

Table 13: **ID/OOD Task Performance (Instance-Based Human Rationales).** This table enlists the ID and OOD performance of different ER criteriona (MSE, MAE, BCE, Huber, Order) and compares them to a setting without ER (None). All models (with or without ER) are trained on the ID dataset and evaluated on the ID and OOD datasets without the need of machine or human rationales. Metrics displayed here (higher the better) for sentiment analysis is Accuracy and Macro F1 for NLI and NER.