UNIREX: A Unified Learning Framework for Language Model Rationale Extraction

Anonymous ACL submission

Abstract

An extractive rationale explains a language model's (LM's) prediction on a given task instance by highlighting the text inputs that most influenced the prediction. Ideally, rationale extraction should be *faithful* and *plausible*, without compromising the LM's (*i.e.,* task model's) *task performance*. Although attribution algorithms and select-predict pipelines are common in rationale extraction, they both rely on certain heuristics that hinder them from satisfying all three desiderata. In light of this, we propose UNIREX, a flexible learning framework which generalizes rationale extractor optimization as follows: (1) specify architecture for a learned rationale extractor; (2) select explainability objectives (*i.e.,* faithfulness and plausibility criteria); and (3) jointly the train task model and rationale extractor on the task using selected objectives. UNIREX enables replacing prior works' heuristic design choices with a generic learned rationale extractor in (1) and optimizing it for all three desiderata in (2)-(3). To facilitate comparison between methods w.r.t. multiple desiderata, we introduce the Normalized Relative Gain (NRG) metric. Across five text classification datasets, UNIREX beats baselines by an average of 32.9% NRG and can even generalize to unseen datasets/tasks.

1 Introduction

Large neural language models (LMs) have yielded state-of-the-art performance on various natural language processing (NLP) tasks (Devlin et al., 2018; Liu et al., 2019). However, LMs' complex reasoning processes are notoriously opaque (Rudin, 2019), posing concerns about the societal implications of using LMs for high-stakes decision-making (Bender et al., 2021). Thus, explaining LMs' behavior is crucial for promoting trust, ethics, and safety in NLP systems (Doshi-Velez and Kim, 2017; Lipton, 2018). Given a LM's (*i.e.,* task model's) predicted label on a text classification instance, an *extractive rationale* is a type of explanation that highlights the tokens that most influenced the model to predict that label (Luo et al., 2021). Ideally, rationale extraction should be *faithful* (Ismail et al., 2021; Jain et al., 2020) and *plausible* (DeYoung et al., 2019), without hurting the LM's *task performance* (DeYoung et al., 2019) (Fig. 1).

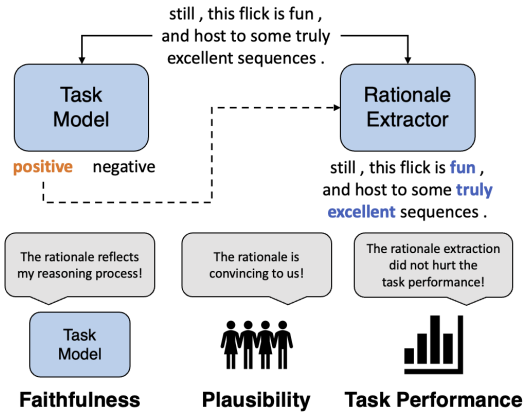

Figure 1: **Desiderata of Rationale Extraction.** Unlike prior works, UNIREX enables optimizing for all three desiderata.

Configuring the rationale extractor and its training can greatly impact these desiderata, yet prior works have commonly adopted two suboptimal heuristics. First, many works rely in some way on *attribution algorithms* (AAs), which extract rationales via handcrafted functions (Sundararajan et al., 2017; Ismail et al., 2021; Situ et al., 2021). AAs cannot be directly trained and tend to be compute-intensive (Bastings and Filippova, 2020). Also, AAs can be a bottleneck for plausibility, as producing human-like rationales is a complex objective requiring high capacity rationale extractors (Narang et al., 2020; DeYoung et al., 2019). Second, many works use a specialized *select-predict pipeline* (SPP), where a predictor module is trained to solve the task using only tokens chosen by a selector module (Jain et al., 2020; Yu et al., 2021; Paranjape et al., 2020). Instead of faithfulness optimization, SPPs heuristically aim for "faithfulness by construction" by treating the selected tokens as a rationale for the predictor's output (which depends

only on those tokens). Still, SPPs typically have worse task performance than vanilla LMs since SPPs hide the full input from the predictor.

To tackle this challenge, we propose the **UNI**fied Learning Framework for **R**ationale **EX**traction (**UNIREX**), which generalizes rationale extractor optimization as follows: (1) specify architecture for a learned rationale extractor; (2) select explainability objectives (*i.e.,* faithfulness and plausibility criteria); and (3) jointly train the task model and rationale extractor on the task using selected objectives (Sec. 3). UNIREX enables replacing prior works' heuristic design choices in (1) with a generic learned rationale extractor and optimizing it for all three desiderata in (2)-(3).

UNIREX provides significant flexibility in performing (1)-(3). For (1), any model architecture is applicable, but we study Transformer LM based rationale extractors in this work (Zaheer et al., 2020; DeYoung et al., 2019). We focus on two architectures: (A) Dual LM, where task model and rationale extractor are separate and (B) Shared LM, where task model and rationale extractor share parameters. For (2), any faithfulness and plausibility criteria can be used. Following DeYoung et al. (2019), we focus on comprehensiveness and sufficiency as faithfulness criteria, while using similarity to gold rationales as plausibility criteria. For (3), trade-offs between the three desiderata can be easily managed during rationale extractor optimization by setting arbitrary loss weights for the faithfulness and plausibility objectives. Plus, though computing the faithfulness criteria involves discrete (non-differentiable) token selection, using Shared LM can approximate end-to-end training and enable both task model and rationale extractor to be optimized w.r.t. all three desiderata (Sec. 3.3).

To evaluate all three desiderata in aggregate, we introduce the Normalized Relative Gain (NRG) metric. Across five text classification datasets – SST, Movies, CoS-E, MultiRC, and e-SNLI (Carton et al., 2020; DeYoung et al., 2019) – our best UNIREX configuration outperforms the strongest baselines by an average of 32.9% NRG (Sec. 4.2), showing that UNIREX can optimize rationale extractors for all three desiderata. In addition, we verify our UNIREX design choices via extensive ablation studies (Sec. 4.3). Furthermore, UNIREX-trained extractors have high generalization power, yielding high plausiblity with minimal gold rationale supervision (Sec. 4.4) and high faithfulness on unseen datasets and tasks (Sec. 4.5). Finally, our user study shows that humans judge UNIREX rationales as more plausible than rationales extracted using other methods (Sec. 4.6).

2 Problem Formulation

Rationale Extraction Let $\mathcal{F}_{\text{task}} = f_{\text{task}}(f_{\text{enc}}(\cdot))$ be a task model for M-class text classification (Sec. A.1), where f_{enc} is the text encoder and f_{task} is the task output head. Typically, $\mathcal{F}_{\text{task}}$ has a BERT-style architecture (Devlin et al., 2018), in which f_{enc} is a Transformer (Vaswani et al., 2017) while f_{task} is 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, and $\mathcal{F}_{\text{task}}(\mathbf{x}_i) \in \mathbb{R}^M$ be the logit vector for the output of the task model. Let $\hat{y}_i = \arg\max_j \mathcal{F}_{\text{task}}(\mathbf{x}_i)_j$ be the class predicted by $\mathcal{F}_{\text{task}}$. Given $\mathcal{F}_{\text{task}}$, $\mathbf{x}_i$, and $\hat{y}_i$, the goal of rationale extraction is to output vector $\mathbf{s}_i = [s_i^t]_{t=1}^n \in \mathbb{R}^n$, such that each $s_i^t \in \mathbb{R}$ is an *importance score* indicating how much token x_i^t influenced $\mathcal{F}_{\text{task}}$ to predict class $\hat{y}_i$. Let $\mathcal{F}_{\text{ext}}$ be a rationale extractor, such that $\mathbf{s}_i = \mathcal{F}_{\text{ext}}(\mathcal{F}_{\text{task}}, \mathbf{x}_i, \hat{y}_i)$. $\mathcal{F}_{\text{ext}}$ can be a learned or heuristic function. In practice, the final rationale is often obtained by binarizing $\mathbf{s}_i$ as $\mathbf{r}_i \in \{0,1\}^n$, via the top-$k\%$ strategy: $r_i^t = 1$ if s_i^t is one of the top-$k\%$ scores in $\mathbf{s}_i$; otherwise, $r_i^t = 0$ (DeYoung et al., 2019; Jain et al., 2020; Pruthi et al., 2020; Chan et al., 2021). For top-$k\%$, let $\mathbf{r}_i^{(k)}$ be the "important" (*i.e.,* ones) tokens in $\mathbf{r}_i$, when using $0 \leq k \leq 100$.

Faithfulness means how well a rationale reflects $\mathcal{F}_{\text{task}}$'s true reasoning process for predicting $\hat{y}_i$ (Jacovi and Goldberg, 2020). Hence, faithfulness metrics measure how much the $\mathbf{r}_i^{(k)}$ tokens impact $p_{\hat{y}_i}(\mathbf{x}_i)$, which denotes $\mathcal{F}_{\text{task}}$'s confidence probability for $\hat{y}_i$ when using $\mathbf{x}_i$ as input (DeYoung et al., 2019; Shrikumar et al., 2017; Hooker et al., 2018; Pruthi et al., 2020). Recently, comprehensiveness and sufficiency have emerged as popular faithfulness metrics (DeYoung et al., 2019). ***Comprehensiveness*** (comp) measures the change in $p_{\hat{y}_i}$ when $\mathbf{r}_i^{(k)}$ is *removed* from the input: $\text{comp} = p_{\hat{y}_i}(\mathbf{x}_i) - p_{\hat{y}_i}(\mathbf{x}_i \backslash \mathbf{r}_i^{(k)})$. ***Sufficiency*** (suff) measures the change in $p_{\hat{y}_i}$ when only $\mathbf{r}_i^{(k)}$ is *kept* in the input: $\text{suff} = p_{\hat{y}_i}(\mathbf{x}_i) - p_{\hat{y}_i}(\mathbf{r}_i^{(k)})$. High faithfulness is signaled by high comp and low suff.

Plausibility means how convincing a rationale is to humans (Jacovi and Goldberg, 2020). This can be measured by automatically computing the similarity between $\mathcal{F}_{\text{ext}}$'s rationales (either $\mathbf{s}_i$ or $\mathbf{r}_i$)

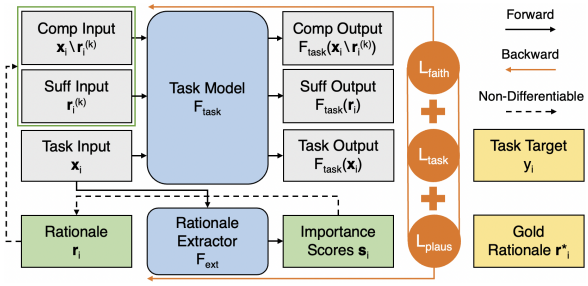

Figure 2: **UNIREX Framework.** UNIREX supports learnable rationale extractors, optimized for faithfulness, plausibility, and task performance.

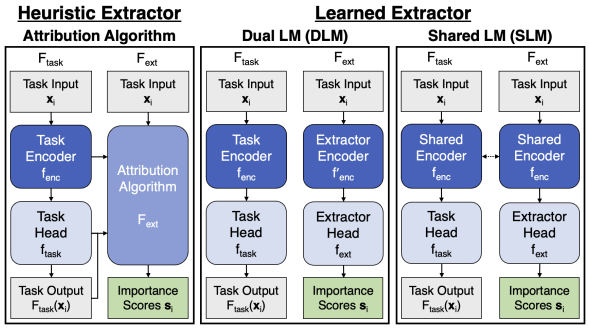

Figure 3: **Rationale Extractor Types.**

and human-annotated gold rationales (DeYoung et al., 2019), or by asking human annotators to rate whether $\mathcal{F}_{\text{ext}}$'s rationales make sense for predicting $\hat{y}_i$ (Strout et al., 2019; Doshi-Velez and Kim, 2017). Typically, a gold rationale is a binary vector $\mathbf{r}_i^* \in \{0,1\}^n$, where ones/zeros indicate important/unimportant tokens (Lei et al., 2016).

Task Performance, w.r.t. rationale extraction, concerns how much $\mathcal{F}_{\text{task}}$'s task performance (on test set) drops when $\mathcal{F}_{\text{task}}$ is trained with explainability objectives (*i.e.,* faithfulness, plausibility) for $\mathcal{F}_{\text{ext}}$. As long as $\mathcal{F}_{\text{task}}$ is trained with non-task losses, $\mathcal{F}_{\text{task}}$'s task performance can be affected.

3 UNIREX

Given task model $\mathcal{F}_{\text{task}}$, UNIREX generalizes rationale extractor optimization as follows: (1) choose architecture for a learned rationale extractor $\mathcal{F}_{\text{ext}}$; (2) select explainability objectives (*i.e.,* faithfulness loss $\mathcal{L}_{\text{faith}}$ and plausibility loss $\mathcal{L}_{\text{plaus}}$); and (3) jointly train $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$ using $\mathcal{L}_{\text{task}}$ (task loss), $\mathcal{L}_{\text{faith}}$, and $\mathcal{L}_{\text{plaus}}$. UNIREX training consists of two backpropagation paths (Fig. 2). The first path is used to update $\mathcal{F}_{\text{task}}$ w.r.t. $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{faith}}$. Whereas $\mathcal{L}_{\text{task}}$ is computed w.r.t. the task target y_i, $\mathcal{L}_{\text{faith}}$ is computed only using the task input $\mathbf{x}_i$ and the top-$k\%$ important tokens $\mathbf{r}_i^{(k)}$ (obtained via $\mathcal{F}_{\text{ext}}$), based on some combination of comp and suff (Sec. 2). The second path is used to update $\mathcal{F}_{\text{ext}}$ w.r.t. $\mathcal{L}_{\text{plaus}}$, which encourages importance scores $\mathbf{s}_i$ to approximate gold rationale $\mathbf{r}_i^*$. Thus, UNIREX frames rationale extraction as the following optimization problem:

$$\begin{aligned}\min_{\mathcal{F}_{\text{task}},\mathcal{F}_{\text{ext}}} \; & \mathcal{L}_{\text{task}}(\mathbf{x}_i, y_i; \mathcal{F}_{\text{task}}) \\ & + \alpha_f \mathcal{L}_{\text{faith}}(\mathbf{x}_i, \mathbf{r}_i^{(k)}; \mathcal{F}_{\text{task}}) \\ & + \alpha_p \mathcal{L}_{\text{plaus}}(\mathbf{x}_i, \mathbf{r}_i^*; \mathcal{F}_{\text{ext}}),\end{aligned} \tag{1}$$

where α_f and α_p are loss weights. If $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$ share parameters, then the shared parameters will be optimized w.r.t. all losses. During inference, for task input $\mathbf{x}_i$, we first use $\mathcal{F}_{\text{task}}$ to predict y_i, then use $\mathcal{F}_{\text{ext}}$ to output a rationale $\mathbf{r}_i$ for $\mathcal{F}_{\text{task}}$'s prediction $\hat{y}_i$. Below, we discuss options for the rationale extractor and explainability objectives.

3.1 Rationale Extractor

In UNIREX, $\mathcal{F}_{\text{ext}}$ is a learned function by default. Learned $\mathcal{F}_{\text{ext}}$ can be any model that transforms x_i^t into s_i^t. Given their success in NLP explainability (DeYoung et al., 2019), we focus on pre-trained Transformer LMs and highlight two architectures: Dual LM (DLM) and Shared LM (SLM) (Fig. 3). For DLM, $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$ are two separate Transformer LMs. DLM provides more dedicated capacity for $\mathcal{F}_{\text{ext}}$, which can help $\mathcal{F}_{\text{ext}}$ output plausible rationales. For SLM, $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$ are two Transformer LMs sharing encoder f_{enc}, while $\mathcal{F}_{\text{ext}}$ has its own output head f_{ext}. SLM leverages multitask learning between $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$, which can improve faithfulness since $\mathcal{F}_{\text{ext}}$ gets more information about $\mathcal{F}_{\text{task}}$'s reasoning process. Unlike heuristic $\mathcal{F}_{\text{ext}}$ (Sec. A.2), learned $\mathcal{F}_{\text{ext}}$ can be optimized for faithfulness/plausibility, but cannot be used out of the box without training. Learned $\mathcal{F}_{\text{ext}}$ is preferred if: (A) optimizing for both faithfulness and plausibility, and (B) gold rationales are available for plausibility optimization (Sec. A.3).

3.2 Explainability Objectives

After selecting $\mathcal{F}_{\text{ext}}$, we specify the explainability objectives, which can be any combination of faithfulness and plausibility criteria. In prior approaches (*e.g.,* AA, SPPs), the rationale extractor is not optimized for both faithfulness and plausibility, but UNIREX makes this possible. For any choice of learned $\mathcal{F}_{\text{ext}}$, UNIREX lets us easily "plug and play" different criteria and loss weights, based on our needs and domain knowledge, to find those that best balance the rationale extraction desiderata.

Faithfulness Evaluating rationale faithfulness is still an open problem with many existing metrics,

and UNIREX is not tailored for any specific metric. Still, given the prevalence of comp/suff (Sec. 2), we focus on comp/suff based objectives.

Recall that comp measures the importance of tokens in $\mathbf{r}_i^{(k)}$ as how $p_{\hat{y}_i}(\hat{\mathbf{x}}_i)$, $\mathcal{F}_{\text{task}}$'s predicted probability for class $\hat{y}_i$, changes when those tokens are removed from $\mathbf{x}_i$. Intuitively, we want $p_{\hat{y}_i}(\hat{\mathbf{x}}_i)$ to be higher than $p_{\hat{y}_i}(\mathbf{x}_i \backslash \mathbf{r}_i^{(k)})$, so higher comp is better. Since comp is defined for a single class' probability rather than the label distribution, we can define the comp loss $\mathcal{L}_{\text{comp}}$ via cross-entropy loss $\mathcal{L}_{\text{CE}}$, as in the following *difference criterion* for $\mathcal{L}_{\text{comp}}$:

$$\begin{aligned}\mathcal{L}_{\text{comp-diff}} &= \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i) \\ &\quad - \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i \backslash \mathbf{r}_i^{(k)}), y_i))\end{aligned} \tag{2}$$

$$\mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i) = -y_i \log(\mathcal{F}_{\text{task}}(\mathbf{x}_i)) \tag{3}$$

For training stability, we compute comp loss for target class y_i here instead of $\mathcal{F}_{\text{task}}$'s predicted class $\hat{y}_i$, since $\hat{y}_i$ is a moving target during training. Using $\mathcal{L}_{\text{comp-diff}}$, it is possible for $\mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i \backslash \mathbf{r}_i^{(k)}), y_i))$ to become much larger than $\mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i)$, leading to arbitrarily negative losses. To avoid this, we can add margin m_c to the loss function, giving the *margin criterion*:

$$\begin{aligned}\mathcal{L}_{\text{comp-margin}} &= \max(-m_c, \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i) \\ &\quad - \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i \backslash \mathbf{r}_i^{(k)}), y_i)) + m_c\end{aligned} \tag{4}$$

Recall that suff measures the importance of tokens in $\mathbf{r}_i^{(k)}$ as how $p_{\hat{y}_i}(\hat{\mathbf{x}}_i)$, $\mathcal{F}_{\text{task}}$'s predicted probability for class $\hat{y}_i$, changes when they are the only tokens kept in $\mathbf{x}_i$. Based on suff's definition, we want $p_{\hat{y}_i}(\mathbf{r}_i^{(k)})$ to be higher than $p_{\hat{y}_i}(\hat{\mathbf{x}}_i)$, so lower suff is better. For suff loss $\mathcal{L}_{\text{suff}}$, we define the difference and margin criteria analogously with margin m_s but the opposite sign (since lower suff is better):

$$\begin{aligned}\mathcal{L}_{\text{suff-diff}} &= \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{r}_i^{(k)}), y_i) \\ &\quad - \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i)\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}_{\text{suff-margin}} &= \max(-m_s, \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{r}_i^{(k)}), y_i) \\ &\quad - \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i)) + m_s\end{aligned} \tag{6}$$

In our experiments, we find that the margin-based comp/suff criteria are effective (Sec. 4.3), though others (*e.g.,* KL Div, MAE) can be used too (Sec. A.4.1). Note that $\mathbf{r}_i^{(k)}$ is computed via top-$k\%$ thresholding (Sec. 2), so we also need to specify a set K of threshold values. We separately compute the comp/suff losses for each $k \in K$, then obtain the final comp/suff losses by averaging over all k values via area-over-precision-curve (AOPC) (DeYoung et al., 2019). To reflect this, we denote the comp and suff losses as $\mathcal{L}_{\text{comp},K}$ and $\mathcal{L}_{\text{suff},K}$, respectively. Let $\alpha_f \mathcal{L}_{\text{faith}} = \alpha_c \mathcal{L}_{\text{comp},K} + \alpha_s \mathcal{L}_{\text{suff},K}$, where α_c and α_s are loss weights.

Plausibility Plausibility is defined as how convincing a rationale is to humans (Jacovi and Goldberg, 2020), *i.e.,* whether humans would agree the rationale supports the model's prediction. While optimizing for plausibility should ideally involve human-in-the-loop feedback, this is prohibitive. Instead, many works consider gold rationales as a cheaper form of plausibility annotation (DeYoung et al., 2019; Narang et al., 2020; Jain et al., 2020). Thus, if gold rationale supervision is available, then we can optimize for plausibility. With gold rationale $\mathbf{r}_i^*$ for input $\mathbf{x}_i$, plausibility optimization entails training $\mathcal{F}_{\text{ext}}$ to predict binary importance label $\mathbf{r}_i^{*,t}$ for each token x_i^t. This is essentially token classification, so one natural choice for $\mathcal{L}_{\text{plaus}}$ is the token-level binary cross-entropy (BCE) criterion:

$$\mathcal{L}_{\text{plaus-BCE}} = -\sum_t \mathbf{r}_i^{*,t} \log(\mathcal{F}_{\text{ext}}(x_i^t)) \tag{7}$$

Besides BCE loss, we can also consider other criteria like sequence-level KL divergence and L1 loss. See Sec. A.4.2 for discussion of these and other plausibility criteria.

3.3 Training and Inference

After setting $\mathcal{F}_{\text{ext}}$, $\mathcal{L}_{\text{faith}}$, and $\mathcal{L}_{\text{plaus}}$, we can move on to training $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$. Since top-$k\%$ rationale binarization (Sec. 3.2) is not differentiable, by default, we cannot backpropagate $\mathcal{L}_{\text{faith}}$ through all of $\mathcal{F}_{\text{ext}}$'s parameters. Thus, $\mathcal{F}_{\text{task}}$ is trained via $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{faith}}$, while $\mathcal{F}_{\text{ext}}$ is only trained via $\mathcal{L}_{\text{plaus}}$. This means $\mathcal{F}_{\text{ext}}$'s rationales $\mathbf{r}_i$ are indirectly optimized for faithfulness by regularizing $\mathcal{F}_{\text{task}}$ such that its behavior aligns with $\mathbf{r}_i$. The exception is if we are using the SLM variant, where encoder f_{enc} is shared by $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$. In this case, f_{enc} is optimized w.r.t. all losses, f_{task} is optimized w.r.t. $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{faith}}$, and f_{ext} is optimized w.r.t. $\mathcal{L}_{\text{plaus}}$. SLM is a simple way to approximate end-to-end training of $\mathcal{F}_{\text{task}}$ and $\mathcal{F}_{\text{ext}}$, whereas past SPPs have used complex methods like reinforcement learning (Lei et al., 2016) and the reparameterization trick (Bastings et al., 2019).

Now, we summarize the full learning objective. Given that cross-entropy loss $\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i)$ is used to train $\mathcal{F}_{\text{task}}$ to predict

y_i, the full learning objective is:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{task}} + \alpha_f \mathcal{L}_{\text{faith}} + \alpha_p \mathcal{L}_{\text{plaus}} \\ &= \mathcal{L}_{\text{task}} + \alpha_c \mathcal{L}_{\text{comp},K} + \alpha_s \mathcal{L}_{\text{suff},K} + \alpha_p \mathcal{L}_{\text{plaus}}.\end{aligned} \tag{8}$$

During inference, we use $\mathcal{F}_{\text{task}}$ to predict y_i, then use $\mathcal{F}_{\text{ext}}$ to output $\mathbf{r}_i$ for $\mathcal{F}_{\text{task}}$'s predicted label $\hat{y}_i$.

4 Experiments

We present empirical results demonstrating UNIREX's effectiveness in managing trade-offs between faithfulness, plausibility, and task performance during rationale extractor optimization. First, our main experiments compare methods w.r.t. faithfulness, plausibility, and task performance (Sec. 4.2). Second, we perform various ablation studies to verify our design choices for UNIREX (Sec. 4.3). Third, we present experiments highlighting UNIREX's generalization ability, both in terms of limited gold rationale supervision (Sec. 4.4) and zero-shot transfer (Sec. 4.5). Fourth, we conduct a user study to further evaluate UNIREX rationales' plausibility, relative to those generated by other methods (Sec. 4.6). See Sec. A.5 for implementation details (LM architecture, AA settings, training).

4.1 Experiment Setup

Datasets We primarily use SST (Socher et al., 2013; Carton et al., 2020), Movies (Zaidan and Eisner, 2008), CoS-E (Rajani et al., 2019), MultiRC (Khashabi et al., 2018), and e-SNLI (Camburu et al., 2018), all of which have gold rationale annotations. The latter four datasets were taken from the ERASER benchmark (DeYoung et al., 2019).

Metrics We use the metrics from the ERASER explainability benchmark (DeYoung et al., 2019). For faithfulness, we use comprehensiveness (Comp) and sufficiency (Suff), for $k = [1, 5, 10, 20, 50]$ (DeYoung et al., 2019). For plausibility, we use area under precision-recall curve (AUPRC) and token F1 (TF1) to measure similarity to gold rationales (DeYoung et al., 2019; Narang et al., 2020). For task performance, we follow (DeYoung et al., 2019) and (Carton et al., 2020) in using accuracy (SST, CoS-E) and macro F1 (Movies, MultiRC, e-SNLI).

To aggregately evaluate multiple desiderata, we introduce the Normalized Relative Gain (NRG) metric, which is based on the ARG metric from Ye et al. (2021). NRG normalizes raw metrics (*e.g.,* F1, sufficiency) to scores between 0 and 1 (higher is better). Given a set of raw metric scores $Z = \{z_1, z_2, ...\}$ (each from a different method), $\text{NRG}(z_i)$ captures z_i's value relative to $\min(Z)$ and $\max(Z)$. If higher values are better for the given metric (*e.g.,* F1), then we have: $\text{NRG}(z_i) = \frac{z_i - \min(Z)}{\max(Z) - \min(Z)}$. If lower values are better (*e.g.,* sufficiency), then we have: $\text{NRG}(z_i) = \frac{\max(Z) - z_i}{\max(Z) - \min(Z)}$. After computing NRG for multiple raw metrics, we can aggregate them w.r.t. desiderata via averaging. Let FNRG, PNRG, and TNRG be the NRG values for faithfulness, plausibility, and task performance, respectively. Finally, we compute the composite NRG as: $\text{CNRG} = \frac{\text{FNRG}+\text{PNRG}+\text{TNRG}}{3}$.

Results Reporting For all results, we report average over three seeds and the five k values. We denote each UNIREX configuration with "([*rationale extractor*]-[*explainability objectives*])". F, P, and FP denote faithfulness, plausibility, and faithfulness+plausibility, respectively.

Baselines The first category is AAs, which are not trained: AA (Grad) (Simonyan et al., 2013), AA (Input*Grad) (Denil et al., 2014), AA (DeepLIFT) (Lundberg and Lee, 2017), AA (IG) (Sundararajan et al., 2017). We also experiment with IG for L2E (Situ et al., 2021), which distills knowledge from an AA to an LM. The second category is SPPs: FRESH (Jain et al., 2020) and A2R (Yu et al., 2021). For FRESH, we use a strong variant where IG rationales are directly given to the predictor, rather than output by a trained selector. A2R aims to improve SPP task performance by regularizing the predictor with an attention-based predictor that uses the full input. In addition, we introduce FRESH+P and A2R+P, which augment FRESH and A2R, respectively, with plausibility optimization. The third category is AA-based regularization: SGT (Ismail et al., 2021), which uses a sufficiency-based criterion to optimize for faithfulness. We also consider SGT+P, which augments SGT with plausibility optimization.

4.2 Main Results

Figs. 4-5 display the main results. For both figures, the left and right plots show methods without and with gold rationales, respectively. In Fig. 4, we compare the CNRG for all methods and datasets. In both plots, we see that UNIREX variants achieve the best CNRG across all datasets, indicating that they are effective in balancing the three desiderata. In particular, UNIREX (DLM-FP) and UNIREX

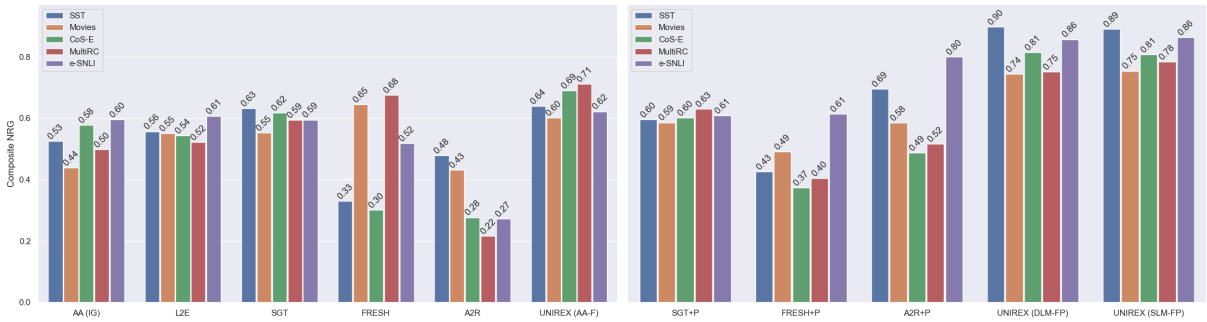

Figure 4: **Composite NRG Comparison.** We show CNRG for all methods and datasets. Left/right plots compare methods with/without gold rationale supervision. Overall, UNIREX variants have the best CNRG.

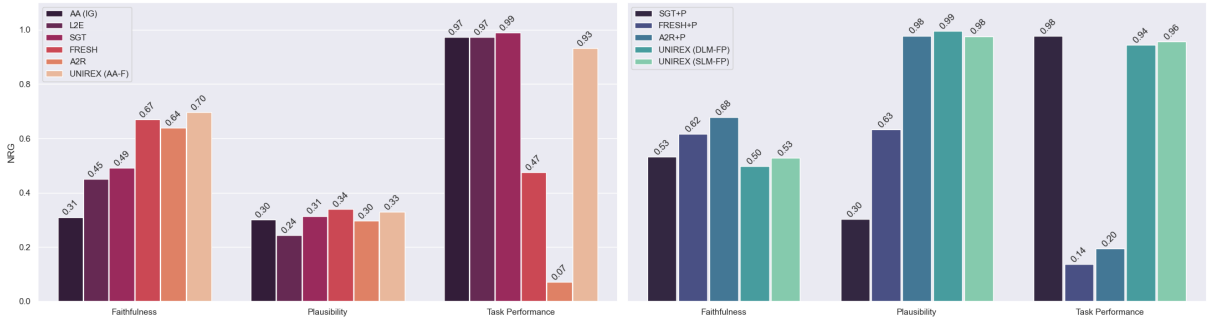

Figure 5: **NRG Comparison by Desiderata.** We show FNRG, PNRG, and TNRG for all methods, averaged over all datasets.

(SLM-FP) have very high CNRG scores, both yielding more than 30% improvement over the strongest baselines. Fig. 5 compares methods w.r.t. per-desiderata NRG (*i.e.,* FNRG, PNRG, TNRG). Again, we see that UNIREX variants achieve a good NRG balance of faithfulness, plausibility, and task performance. Meanwhile, many baselines (*e.g.,* AA (IG), A2R, SGT+P) do well on some desiderata but very poorly on others.

4.3 Ablation Studies

We present five ablation studies to validate the effectiveness of our UNIREX design choices. The ablation results are displayed in Table 1. In this table, each of the five sections shows results for a different ablation. Thus, all numbers within the same section and column are comparable.

Extractor Type In the Ext Type (F) section, we compare four heuristic rationale extractors, using AA-F. Rand uses random importance scores, Gold directly uses the gold rationales, Inv uses the inverse of the gold rationales, and IG uses IG. All heuristics yield similar task performance, but IG dominates on all faithfulness metrics. This makes sense because IG is computed using $\mathcal{F}_{\text{task}}$'s inputs/parameters/outputs, while the others do not have this information. For plausibility, Gold is the best, Inv is the worst, and Rand and IG are about the same, as none of the heuristics are optimized

Ablation	UNIREX Config	Faithfulness		Plausibility	Performance
		Comp (↑)	Suff (↓)	AUPRC (↑)	Acc (↑)
Ext Type (F)	AA-F (Rand)	0.171 (±0.040)	0.327 (±0.050)	44.92 (±0.00)	94.05 (±0.35)
	AA-F (Gold)	0.232 (±0.088)	0.249 (±0.021)	100.00 (±0.00)	93.81 (±0.54)
	AA-F (Inv)	0.242 (±0.010)	0.357 (±0.019)	20.49 (±0.00)	93.47 (±1.81)
	AA-F (IG)	0.292 (±0.051)	0.171 (±0.038)	48.13 (±1.14)	92.97 (±0.44)
Ext Type (FP)	AA-FP (Sum)	0.296 (±0.067)	0.185 (±0.048)	47.60 (±2.44)	93.25 (±0.45)
	AA-FP (MLP)	0.285 (±0.051)	0.197 (±0.100)	54.82 (±1.97)	93.23 (±0.92)
	DLM-FP	0.319 (±0.090)	0.167 (±0.036)	85.80 (±0.74)	93.81 (±0.18)
	SLM-FP	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	93.68 (±0.67)
Comp/Suff Loss	SLM-FP (Comp)	0.350 (±0.048)	0.310 (±0.049)	82.79 (±0.62)	93.59 (±0.11)
	SLM-FP (Suff)	0.166 (±0.003)	0.152 (±0.012)	83.74 (±0.84)	94.16 (±0.39)
	SLM-FP (Comp+Suff)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	93.68 (±0.67)
Suff Criterion	SLM-FP (KL Div)	0.306 (±0.098)	0.131 (±0.005)	82.62 (±0.88)	93.06 (±0.25)
	SLM-FP (MAE)	0.278 (±0.058)	0.143 (±0.008)	82.66 (±0.61)	93.78 (±0.13)
	SLM-FP (Margin)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	93.68 (±0.67)
SLM Ext Head	SLM-FP (Linear)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	93.68 (±0.67)
	SLM-FP (MLP-2048-2)	0.323 (±0.071)	0.144 (±0.012)	83.82 (±0.77)	93.67 (±0.18)
	SLM-FP (MLP-4096-3)	0.295 (±0.057)	0.154 (±0.027)	84.53 (±0.61)	93.19 (±0.79)

Table 1: **UNIREX Ablation Studies on SST.**

for plausibility. In the Ext Type (FP) section, we compare four learned rationale extractors. By default, attribution algorithms' dimension scores are pooled into token scores via sum pooling. AA-FP (Sum) uses IG with sum pooling, while AA-FP (MLP) replaces the sum pooler with a MLP-based pooler to increase capacity for plausibility optimization. Task performance for all four methods is similar, AA-FP (Sum) dominates on faithfulness, and DLM-FP and SLM-FP dominate on plausibility. AA-FP (MLP) does not perform as well on faithfulness but slightly improves on plausibility compared to AA-FP (Sum).

Comp/Suff Losses The Comp/Suff Loss section compares different combinations of Comp and Suff losses, using SLM-FP. Note that SLM-FP (Comp+Suff) is equivalent to SLM-FP shown

in other tables/sections. As expected, SLM-FP (Comp) does best on Comp, but SLM-FP (Comp+Suff) actually does best on Suff. Meanwhile, SLM-FP, (Suff) does second-best on Suff but is much worse on Comp. This shows that Comp and Suff are complementary for optimization.

Suff Criterion The Suff Criterion section compares different Suff criteria, using SLM-FP. SLM-FP (KLDiv) uses the KL divergence criterion, SLM-FP (MAE) uses the MAE criterion, and SLM-FP (Margin) uses the margin criterion. SLM-FP (Margin) is equivalent to SLM-FP in other tables/sections. All criteria yield similar performance and plausibility, while Margin is slightly better on faithfulness.

SLM Extractor Head The SLM Ext Head section compares different extractor heads, using SLM-FP. Linear is the default choice and uses a linear layer. MLP-2048-2 uses a MLP with two 2048-dim hidden layers. MLP-4096-3 uses a MLP with three 4096-dim hidden layers. All three output head types yield similar performance, but decreasing head capacity yields better faithfulness, while increasing head capacity heads yields better plausibility. This trades off faithfulness and plausibility, although larger heads will be more compute-intensive.

4.4 Gold Rationale Data Efficiency

UNIREX supports arbitrary amounts of gold rationale supervision and allows us to account for data efficiency. In Fig. 6, we compare plausibility (in AUPRC) for $\gamma = [0.5, 1, 5, 10, 20, 100]$ (*i.e.,* % of train instances with gold rationales). We compare AA (IG) and four UNIREX variants (AA-F, AA-FP, DLM-FP, SLM-FP). AA (IG) and AA-F do not use gold rationales and thus have the same AUPRC for all γ. Standard deviation is shown by the error bands. UNIREX (DLM-FP) and UNIREX (SLM-FP) dominate across all γ values, with AUPRC slowly decreasing as γ decreases. Even at $\gamma = 0.5$, they can still achieve high AUPRC. This suggests that UNIREX's gold rationale batching procedure (Sec. A.3) is effective for learning from minimal gold rationale supervision and demonstrates how UNIREX enables us to manage this trade-off. See Sec. A.6 for similar results on CoS-E.

4.5 Zero-Shot Explainability Transfer

In Table 10, we investigate if UNIREX-trained models can generalize to unseen datasets and tasks in a zero-shot setting (*i.e.,* no fine-tuning on target datasets). Plausibility is not evaluated here, since these unseen datasets do not have gold rationales. As the source model, we compare various SST-trained models: AA (IG) and UNIREX (AA-F, DLM-FP). First, we evaluate on unseen datasets for a seen task (sentiment analysis (SA)): Yelp (Zhang et al., 2015) and Amazon (McAuley and Leskovec, 2013). Second, we evaluate on unseen datasets for unseen tasks: Stormfront (hate speech detection (HSD), binary F1) (de Gibert et al., 2018), OffenseEval (offensive speech detection (OSD), macro F1) (Zampieri et al., 2019), and SemEval2018 (irony detection (ID), binary F1) (Van Hee et al., 2018). In both settings, DLM-FP has the best balance of faithfulness and task performance. AA-F is not as good as DLM-FP, but still always beats AA (IG) on faithfulness. This shows that UNIREX-trained models' faithfulness and task performance are both transferable to unseen datasets/tasks.

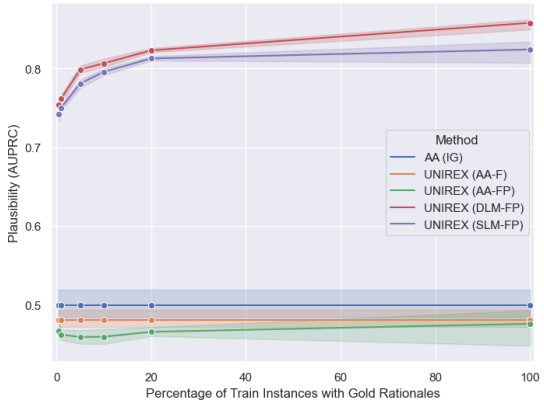

Figure 6: **Gold Rationale Data Efficiency on SST.**

Task	Dataset	Method	Faithfulness		Task Performance
			Comp (↑)	Suff (↓)	Perf (↑)
SA	SST	AA (IG)	0.119 (±0.009)	0.258 (±0.031)	93.81 (±0.55)
		UNIREX (AA-F)	0.292 (±0.051)	0.171 (±0.038)	92.97 (±0.44)
		UNIREX (DLM-FP)	0.319 (±0.090)	0.167 (±0.036)	93.81 (±0.54)
	Yelp	AA (IG)	0.069 (±0.004)	0.219 (±0.028)	92.50 (±2.07)
		UNIREX (AA-F)	0.138 (±0.078)	0.126 (±0.059)	83.93 (±13.20)
		UNIREX (DLM-FP)	0.265 (±0.094)	0.097 (±0.033)	92.37 (±0.46)
	Amazon	AA (IG)	0.076 (±0.010)	0.224 (±0.037)	91.13 (±0.28)
		UNIREX (AA-F)	0.130 (±0.077)	0.073 (±0.039)	77.90 (±13.12)
		UNIREX (DLM-FP)	0.232 (±0.072)	0.098 (±0.033)	89.35 (±2.22)
HSD	Stormfront	AA (IG)	0.135 (±0.010)	0.245 (±0.059)	10.48 (±1.66)
		UNIREX (AA-F)	0.219 (±0.009)	0.092 (±0.025)	10.36 (±1.94)
		UNIREX (DLM-FP)	0.167 (±0.084)	0.115 (±0.059)	10.37 (±2.66)
OSD	OffenseEval	AA (IG)	0.097 (±0.009)	0.244 (±0.052)	33.51 (±0.99)
		UNIREX (AA-F)	0.074 (±0.040)	0.102 (±0.024)	32.62 (±4.85)
		UNIREX (DLM-FP)	0.140 (±0.049)	0.087 (±0.045)	35.52 (±1.26)
ID	SemEval2018	AA (IG)	0.128 (±0.014)	0.248 (±0.064)	29.63 (±4.72)
		UNIREX (AA-F)	0.069 (±0.041)	0.096 (±0.011)	49.95 (±8.31)
		UNIREX (DLM-FP)	0.149 (±0.052)	0.102 (±0.053)	31.97 (±2.80)

Table 2: **Zero-Shot Explainability Transfer from SST.**

4.6 User Study on Plausibility

Gold rationale based plausibility evaluation is noisy because gold rationales are for the target label, not a model's predicted label. Thus, we conduct two

Method	Forward Simulation		Subjective Rating
	Accuracy (%)	Confidence (1-4)	Alignment (1-5)
No Rationale	0.92 (±0.05)	3.02 (±0.39)	-
SGT+P	0.81 (±0.10)	2.34 (±0.37)	3.64 (±0.29)
A2R+P	0.41 (±0.17)	2.83 (±0.31)	2.97 (±0.56)
UNIREX (AA-FP)	0.72 (±0.11)	2.00 (±0.36)	3.26 (±0.40)
UNIREX (DLM-FP)	0.84 (±0.07)	2.77 (±0.34)	3.96 (±0.28)
Gold	0.81 (±0.04)	2.88 (±0.31)	4.00 (±0.26)

Table 3: **Plausibility User Study on SST.**

five-annotator user studies (Table 3) to get a better plausibility measurement. Given 50 random test instances from SST, we get the rationales for SGT+P, A2R+P, UNIREX (AA-FP), and UNIREX (DLM-FP), plus the gold rationales. For each instance, we threshold all rationales to have the same number of positive tokens as the gold rationale. The first user study is forward simulation (Hase and Bansal, 2020; Jain et al., 2020). Here, the annotator is given an input and a rationale for some model's prediction, then asked what (binary) sentiment label the model most likely predicted. For forward simulation, we also consider a No Rationale baseline, where no tokens are highlighted. For No Rationale and Gold, the target label is the correct choice. Annotators are also asked to rate their confidence (4-point Likert scale) in their answer to this question. The second user study involves giving a subjective rating of how plausible the rationale is (Hase and Bansal, 2020). Here, the annotator is given the input, rationale, and model's predicted label, then asked to rate (5-point Likert scale) how aligned the rationale is with the prediction.

In both forward simulation and subjective rating, we find that DLM-FP performs best among all non-oracle methods and even beats Gold on accuracy, further supporting that DLM-FP rationales are plausible. Meanwhile, SGT+P and AA-FP, which had lower AUPRC/TF1 in our automatic evaluation, also do worse in accuracy/alignment. Also, users found SGT+P and AA-FP rationales harder to understand, as shown by their lower confidence scores. Meanwhile, A2R+P had high AUPRC/TF1, but gets very low accuracy/alignment because A2R+P's predicted label often not the target label, leading to misalignment with its gold-like rationale. A2R+P is a great example of how automatic plausibility evaluation can be misleading. For the accuracy, confidence, and alignment questions, we achieved Fleiss' Kappa (Fleiss, 1971) inter-annotator agreement scores of 0.2456 (fair), 0.1282 (slight), and, 0.1561 (slight), respectively. This lack of agreement shows the difficulty of measuring plausibility.

5 Related Work

Faithfulness Many prior works have tried to improve the faithfulness of extractive rationales through the use of AAs (Bastings and Filippova, 2020). Typically, this involves designing gradient-based (Sundararajan et al., 2017; Denil et al., 2014; Lundberg and Lee, 2017; Li et al., 2015) or perturbation-based (Li et al., 2016; Poerner et al., 2018; Kádár et al., 2017) AAs. However, attribution algorithms cannot be optimized and tend to be compute-intensive. Recently, Ismail et al. (2021) addressed the optimization issue by regularizing the task model to yield faithful rationales via the AA, while Situ et al. (2021) addressed the compute issue by training a (less compute-intensive) LM to mimic an AA's behavior. Another line of work aims to produce faithful rationales by construction, via SPPs (Jain et al., 2020; Yu et al., 2021; Paranjape et al., 2020; Bastings et al., 2019; Lei et al., 2016). Still, SPPs' faithfulness can only guarantee sufficiency – not comprehensiveness (DeYoung et al., 2019). Also, SPPs generally perform worse than vanilla LMs because they hide much of the original text input from the predictor and are hard to train end-to-end.

Plausibility Existing approaches for improving extractive rationale plausibility typically involve supervising LM-based extractors (Bhat et al., 2021) or SPPs (Jain et al., 2020; Paranjape et al., 2020; DeYoung et al., 2019) with gold rationales. However, LM-based extractors are seldom trained for faithfulness, while SPPs' faithfulness comes at the great cost of performance. Meanwhile, more works focus on improving the plausibility of free-text rationales (Narang et al., 2020; Lakhotia et al., 2020; Camburu et al., 2018), often with task-specific pipelines (Rajani et al., 2019; Kumar and Talukdar, 2020).

Connection to UNIREX Unlike prior works, UNIREX enables both the task model and rationale extractor to be jointly optimized for faithfulness, plausibility, and task performance. As a result, UNIREX-trained rationale extractors achieve a better balance of faithfulness and plausibility, without compromising the task model's performance. Also, by using a learned rationale extractor, UNIREX does not have the computational expenses that limit many AAs.

References

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. *arXiv preprint arXiv:1905.08160*.

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*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Meghana Moorthy Bhat, Alessandro Sordoni, and Subhabrata Mukherjee. 2021. Self-training with few-shot rationalization: Teacher explanations aid student in few-shot nlu. *arXiv preprint arXiv:2109.08259*.

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.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

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*.

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

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2018. A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758*.

Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. 2021. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*.

Kushal Lakhotia, Bhargavi Paranjape, Asish Ghoshal, Wen-tau Yih, Yashar Mehdad, and Srinivasan Iyer. 2020. Fid-ex: Improving sequence-to-sequence models for extractive rationale generation. *arXiv preprint arXiv:2012.15482*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

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

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.

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.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. *arXiv preprint arXiv:2005.00652*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. Evaluating neural network explanation methods using hybrid documents and morphological agreement. *arXiv preprint arXiv:1801.06422*.

Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. 2020. Evaluating explanations: How much do explanations from the teacher aid students? *arXiv preprint arXiv:2012.00893*.

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

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.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5340–5355.

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.

Julia Strout, Ye Zhang, and Raymond J Mooney. 2019. Do human rationales improve machine explanations? *arXiv preprint arXiv:1905.13714*.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.

Mo Yu, Yang Zhang, Shiyu Chang, and Tommi Jaakkola. 2021. Understanding interlocking dynamics of cooperative rationalization. *Advances in Neural Information Processing Systems*, 34.

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.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

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

A Appendix

A.1 Text Classification

Here, we formalize the text classification problem in more detail. Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}_{i=1}^{N}$ be a dataset, where $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ are the text inputs, $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ are the labels, and N is the number of instances $(\mathbf{x}_i, y_i)$ in $\mathcal{D}$. We also assume $\mathcal{D}$ can be partitioned into train set $\mathcal{D}_{\text{train}}$, dev set $\mathcal{D}_{\text{dev}}$, and test set $\mathcal{D}_{\text{test}}$. Let $\mathcal{F}_{\text{task}} = f_{\text{task}}(f_{\text{enc}}(\cdot))$ be a task LM, where f_{enc} is the text encoder, and f_{task} is the task output head. Typically, $\mathcal{F}_{\text{task}}$ has a BERT-style architecture (Devlin et al., 2018), in which f_{enc} is a Transformer (Vaswani et al., 2017) while f_{task} is a linear layer. Below, we define the sequence classification (SST, Movies, MultiRC, e-SNLI) and multi-choice QA (CoS-E) tasks, which are different types of text classification.

Sequence Classification In sequence classification, $\mathbf{x}_i$ is a token sequence (*e.g.,* a single sentence, a pair of sentences), while y_i is the target class for $\mathbf{x}_i$. Here, we assume a fixed label space $Y = \{1, ..., M\}$ of size M, where $y_i \in Y$ for all i. Thus, f_{task} outputs a vector of size M, such that $\mathcal{F}_{\text{task}}(\mathbf{x}_i) = f_{\text{task}}(f_{\text{enc}}(\mathbf{x}_i)) = \hat{\mathbf{y}}_i \in \mathbb{R}^M$ is the logit vector used to classify $\mathbf{x}_i$. Given $\hat{\mathbf{y}}_i = [\hat{y}_{i,j}]_{j=1}^{M}$, let $\hat{y}_i = \arg\max_j \hat{y}_{i,j}$ be the class predicted by $\mathcal{F}_{\text{task}}$. The goal of sequence classification is to learn $\mathcal{F}_{\text{task}}$ such that $y_i = \hat{y}_i$, for all $(\mathbf{x}_i, y_i)$ (Minaee et al., 2021).

Multi-Choice QA Instead of a fixed label space, multi-choice QA has a different (but fixed-size) set of answer choices per instance. For instance i, let q_i be the question (*e.g., "A friend is greeting me, what would they say?"*) and $A_i = \{a_{i,j}\}_{j=1}^{M}$ be the corresponding answer choices (*e.g.,* {*"say hello"*, *"greet"*, *"associate"*, *"socialize"*, *"smile"*}), where M is now the number of answer choices. Define $\mathbf{x}_{i,j} = q_i \oplus a_{i,j}$, where $\oplus$ denotes concatenation. In multi-choice QA, we have $\mathbf{x}_i = \{\mathbf{x}_{i,j}\}_{j=1}^{M}$, while $y_i \in A_i$ is the correct answer for $\mathbf{x}_i$. Thus, f_{task} outputs a scalar, such that $\mathcal{F}_{\text{task}}(\mathbf{x}_{i,j}) = f_{\text{task}}(f_{\text{enc}}(\mathbf{x}_{i,j})) = \hat{y}_{i,j} \in \mathbb{R}$ is the logit for $\mathbf{x}_{i,j}$. Given $\hat{\mathbf{y}}_i = [\hat{y}_{i,j}]_{j=1}^{M}$, let $j' = \arg\max_j \hat{y}_{i,j}$, where $\hat{y}_i = a_{i,j'}$ is the answer predicted by $\mathcal{F}_{\text{task}}$. The goal of multi-choice QA is to learn $\mathcal{F}_{\text{task}}$ such that $y_i = \hat{y}_i$, for all $(\mathbf{x}_i, y_i)$ (Talmor et al., 2018).

A.2 Heuristic Rationale Extractors

A heuristic $\mathcal{F}_{\text{task}}$ is an AA, which can be any hand-crafted function that calculates an importance score s_i^t for each input token x_i^t (Bastings and Filippova, 2020). AAs are typically gradient-based (Sundararajan et al., 2017; Denil et al., 2014; Lundberg and Lee, 2017; Li et al., 2015) or perturbation-based (Li et al., 2016; Poerner et al., 2018; Kádár et al., 2017) methods. Gradient-based methods compute s_i^t via the gradient of $\mathcal{F}_{\text{task}}$'s output $\hat{\mathbf{y}}_i$ w.r.t. x_i^t, via one or more $\mathcal{F}_{\text{task}}$ backward passes. Perturbation-based methods measure s_i^t as $\hat{\mathbf{y}}_i$'s change when perturbing (*e.g.,* removing) x_i^t, via multiple $\mathcal{F}_{\text{task}}$ forward passes.

AAs can be used out of the box without training and are designed to satisfy certain faithfulness-related axiomatic properties (Sundararajan et al., 2017; Lundberg and Lee, 2017). However, AAs' lack of learnable parameters means they cannot be optimized for faithfulness/plausibility. Thus, if $\mathcal{F}_{\text{task}}$ is trained for explainability using AA-based rationales, then only $\mathcal{F}_{\text{task}}$ is optimized. Also, faithful AAs tend to be compute-intensive, requiring many $\mathcal{F}_{\text{task}}$ backward/forward passes per instance (Sundararajan et al., 2017; Lundberg and Lee, 2017; Li et al., 2016).

A.3 Gold Rationale Supervision

If a learned rationale extractor is chosen, UNIREX enables users to specify how much gold rationale supervision to use. Ideally, each train instance would be annotated with a gold rationale. In this case, we could directly minimize the plausibility loss for each train instance. However, since gold rationales can be expensive to annotate, UNIREX provides a special batching procedure for training with limited gold rationale supervision.

Given $N_{\text{train}} = |\mathcal{D}_{\text{train}}|$ train instances, let $0 < \gamma < 100$ be the percentage of train instances with gold rationales, $N_{\text{gold}} = \lceil \frac{\gamma}{100} N_{\text{train}} \rceil \geq 1$ be the number of train instances with gold rationales, b be the desired train batch size, and $\beta > 1$ be a scaling factor. Define $\mathcal{D}_{\text{gold}} \subseteq \mathcal{D}_{\text{train}}$ as the set of train instances with gold rationales, where $|\mathcal{D}_{\text{gold}}| = N_{\text{gold}}$. Note that, if all train instances have gold rationales, then $\mathcal{D}_{\text{gold}} = \mathcal{D}_{\text{train}}$ and $\gamma = 100$.

Each batch is constructed as follows: (1) randomly sample $b_{\text{gold}} = \max(1, \frac{b}{\beta})$ instances from $\mathcal{D}_{\text{gold}}$ without replacement, then (2) randomly sample $b - b_{\text{gold}}$ instances from $\mathcal{D}_{\text{train}} \backslash \mathcal{D}_{\text{gold}}$ without replacement. This results in a batch with b total train instances, b_{gold} with gold rationales and the rest without. Since N_{gold} is generally small, we only sample from $\mathcal{D}_{\text{gold}}$ without replacement for a given batch, but not a given epoch. Thus, instances from $\mathcal{D}_{\text{gold}}$ may appear more than once in the same epoch. However, we do sample from $\mathcal{D}_{\text{train}} \backslash \mathcal{D}_{\text{gold}}$ without replacement for each batch and epoch, so every instance in $\mathcal{D}_{\text{train}} \backslash \mathcal{D}_{\text{gold}}$ appears exactly once per epoch.

After constructing the batch, we compute the plausibility loss for the batch as follows: $\sum_{i=1}^{b} \mathbb{1}_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{gold}}} \quad \mathcal{L}_{\text{plaus}}(\mathcal{F}_{\text{ext}}(\mathbf{x}_i), \mathbf{r}_i^*)$, where $\mathcal{L}_{\text{plaus}}$ is the plausibility loss for train instance $(\mathbf{x}_i, y_i)$. This function zeroes out the plausibility loss for instances without gold rationales, so that plausibility is only being optimized with respect to instances with gold rationales. However, in Sec. A.6, we show that it is possible to achieve high plausibility via rationale extractors trained on minimal gold rationale supervision.

A.4 Explainability Objectives

A.4.1 Faithfulness

Sufficiency In addition, to the criteria presented in Sec. 3.2, we consider two other sufficiency loss functions. The first is the *KL divergence criterion* used in (Ismail et al., 2021), which considers the entire label distribution and is defined as $\mathcal{L}_{\text{suff}} = \text{KL}(\mathcal{F}_{\text{task}}(\mathbf{r}_{ik+}) \mid\mid \mathcal{F}_{\text{task}}(\mathbf{x}_i))$. The second is the *MAE criterion*, which is defined as $\mathcal{L}_{\text{suff}} = |\mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{r}_{ik+}), y_i) - \mathcal{L}_{\text{CE}}(\mathcal{F}_{\text{task}}(\mathbf{x}_i), y_i)|$. Unlike the margin criterion, the MAE criterion assumes that using $\mathbf{r}_{ik+}$ as input should not yield better performance than using $\mathbf{x}_i$) as input. In our experiments, we find that the margin-based suff criterion is effective, though others (*e.g.,* KL divergence, MAE) can be used too.

Rationale Thresholding Faithfulness losses are based on rationales binarized using some top-$k\%$ threshold, so we also need to specify a set K of threshold values. For each $k \in K$, we separately compute the comprehensiveness and sufficiency losses. Then, we obtain the final comprehensiveness and sufficiency loss values by averaging over all k values via area-over-precision-curve (AOPC) (DeYoung et al., 2019). To reflect this, we denote the comprehensiveness and sufficiency losses as $\mathcal{L}_{\text{comp},K}$ and $\mathcal{L}_{\text{suff},K}$, respectively. To reflect this, we denote the comp and suff losses as $\mathcal{L}_{\text{comp},K}$ and $\mathcal{L}_{\text{suff},K}$, respectively. Let $\alpha_f \mathcal{L}_{\text{faith}} = \alpha_c \mathcal{L}_{\text{comp},K} + \alpha_s \mathcal{L}_{\text{suff},K}$, where α_c and

α_s are loss weights.

A.4.2 Plausibility

Similar to faithfulness, UNIREX places no restrictions on the choice of plausibility objective. Rationale extraction is essentially a token classification task, so token classification losses are most applicable here. One natural choice is the token-level *binary cross-entropy criterion*, which is denoted as $\mathcal{L}_{\text{plaus}} = \mathcal{L}_{\text{BCE}}(\mathcal{F}_{\text{ext}}(\mathbf{x}_i), \mathbf{r}_i^*)$. Another option is the sequence-level *KL divergence criterion*, which is denoted as $\mathcal{L}_{\text{plaus}} = \text{KL}(\mathcal{F}_{\text{ext}}(\mathbf{x}_i) \mid\mid \mathbf{r}_i^*)$.

Additionally, we can directly penalize $\mathcal{F}_{\text{ext}}(\mathbf{x}_i)$ in the logit space via a *linear loss*, defined as $\mathcal{L}_{\text{plaus}} = \Phi(\mathbf{r}_i^*)\ \mathcal{F}_{\text{ext}}(\mathbf{x}_i)$, where $\Phi(u) = -2u + 1$ maps positive/negative tokens to $-1/+1$. Note that this loss directly pushes the logits corresponding to positive/negative tokens to be higher/lower and increase the margin between them. To prevent linear loss values from becoming arbitrarily negative, we can also use a *margin criterion* here: $\mathcal{L}_{\text{plaus}} = \max(-m_p, \Phi(\mathbf{r}_i^*)\ \mathcal{F}_{\text{ext}}(\mathbf{x}_i)) + m_p$.

A.5 Implementation Details

LM Architecture While many prior works use BERT (Devlin et al., 2018) Transformer LMs, BERT is limited to having sequences with up to 512 tokens, which is problematic since many datasets (*e.g.,* Movies) contain much longer sequences. Meanwhile, BigBird (Zaheer et al., 2020) is a state-of-the-art Transformer LM designed to handle long input sequences with up to 4096 tokens. Thus, we use BigBird-Base, which is initialized with RoBERTa-Base (Liu et al., 2019), in all of our experiments (*i.e.,* both baselines and UNIREX). We obtain the pre-trained BigBird-Base model from the Hugging Face Transformers library (Wolf et al., 2019). Note that UNIREX is agnostic to the choice of LM architecture, so RNNs, CNNs, and other Transformer LMs are also supported by UNIREX. However, we leave exploration of other LM architectures for future work.

Attribution Algorithm In all of our experiments, we use the Integrated Gradients (IG) (Sundararajan et al., 2017) attribution algorithm, which is well-known for being one of the most faithful attribution algorithms and a standard out-of-the-box choice. By default, IG requires 50 steps (*i.e.,* backward passes) per instance, which is prohibitive when regularizing the LM via IG. Thus, for all experiments we use 3 steps in both training and evaluation. Also, we implement IG using the Captum library (Kokhlikyan et al., 2020).

Training For all experiments, we use a learning rate of $2\text{e}{-5}$ and effective batch size of 32. We train for a maximum of 10 epochs, with early stopping patience of 5 epochs. We only tune the faithfulness loss weights, sweeping over loss weights $\alpha_c = [0.5, 0.7, 1.0]$ and $\alpha_s = [0.5, 0.7, 1.0]$. However, we find that $\alpha_c = 0.5$ and $\alpha_s = 0.5$ are usually best. For the plausibility loss weight, we use $\alpha_p = 0.5$. For the batching factor β (Sec. A.3), we use 2. For model selection, we choose the model with the best dev performance averaged over three seeds. We can also perform model selection based on dev explainability metrics, but we leave this extended tuning for future work. We implement the training framework in PyTorch-Lightning (Paszke et al., 2019; Falcon and The PyTorch Lightning team, 2019).

A.6 Gold Rationale Data Efficiency

Fig. 7 shows the gold rationale data efficiency results for CoS-E, using the same setup described in Sec. 4.4. Overall, we see that the CoS-E results are quite similar to the SST results. Again, UNIREX (DLM-FP) and UNIREX (SLM-FP) dominate across all γ values, with AUPRC slowly decreasing as γ decreases. Interestingly, UNIREX (AA-FP) yields a noticeable dip in AUPRC for lower γ values. Since AA-FP has limited capacity (via the task model) for plausibility optimization, it is possible that this fluctuation is due to random noise. We leave further analysis of this to future work.

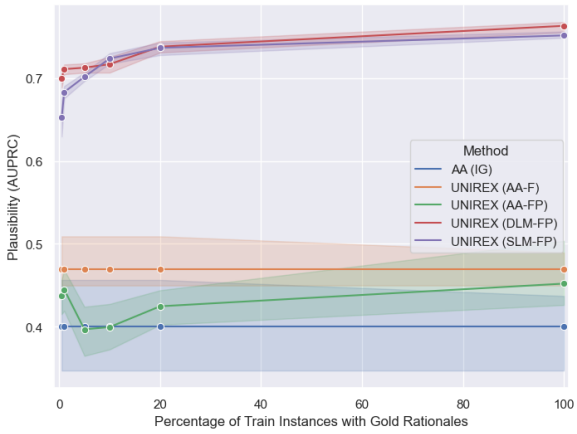

Figure 7: **Gold Rationale Data Efficiency on CoS-E.**

A.7 Raw Empirical Results

In this section, we present more detailed results from our experiments. In particular, Tables 4-10 contain the results using raw faithfulness, plausibility, and performance metrics, in addition to the aggregate NRG metrics presented in the main text.

Method	Composite	Faithfulness			Plausibility			Performance	
	NRG (↑)	NRG (↑)	Comp (↑)	Suff (↓)	NRG (↑)	AUPRC (↑)	TF1 (↑)	NRG (↑)	Acc (↑)
AA (Grad)	0.488	0.337	0.142 (±0.010)	0.256 (±0.006)	0.192	58.86 (±3.65)	27.40 (±0.00)	0.935	93.81 (±0.55)
AA (Input*Grad)	0.420	0.107	0.078 (±0.013)	0.342 (±0.014)	0.218	44.16 (±1.43)	45.02 (±0.39)	0.935	93.81 (±0.55)
AA (DeepLIFT)	0.453	0.122	0.085 (±0.006)	0.340 (±0.018)	0.302	46.50 (±1.32)	50.18 (±0.32)	0.935	93.81 (±0.55)
AA (IG)	0.526	0.297	0.119 (±0.009)	0.258 (±0.031)	0.347	49.94 (±1.77)	50.75 (±0.54)	0.935	93.81 (±0.55)
L2E	0.557	0.487	0.012 (±0.004)	0.009 (±0.024)	0.250	44.84 (±0.32)	47.24 (±0.87)	0.935	93.81 (±0.55)
SGT	0.632	0.555	0.147 (±0.024)	0.113 (±0.031)	0.371	51.38 (±2.47)	51.35 (±1.64)	0.971	94.40 (±0.57)
FRESH	0.330	0.837	0.219 (±0.057)	0.000 (±0.000)	0.152	42.06 (±8.84)	41.19 (±4.01)	0.000	78.78 (±6.48)
A2R	0.479	0.941	0.283 (±0.104)	0.000 (±0.000)	0.457	63.36 (±6.01)	46.74 (±6.65)	0.038	79.39 (±11.67)
UNIREX (AA-F)	0.639	0.706	0.292 (±0.051)	0.171 (±0.038)	0.329	48.13 (±1.14)	50.96 (±0.93)	0.882	92.97 (±0.44)
SGT+P	0.596	0.507	0.139 (±0.032)	0.137 (±0.026)	0.355	50.38 (±1.45)	50.98 (±0.46)	0.928	93.70 (±0.88)
FRESH+P	0.426	0.765	0.175 (±0.043)	0.000 (±0.000)	0.503	60.87 (±9.83)	53.55 (±8.27)	0.011	78.95 (±5.18)
A2R+P	0.695	0.953	0.290 (±0.016)	0.000 (±0.000)	0.978	85.56 (±1.01)	70.97 (±1.03)	0.154	81.26 (±0.52)
UNIREX (DLM-P)	0.770	0.339	0.142 (±0.008)	0.255 (±0.007)	0.970	84.35 (±0.87)	71.54 (±0.53)	1.000	94.86 (±0.41)
UNIREX (AA-FP)	0.636	0.339	0.296 (±0.067)	0.185 (±0.048)	0.315	47.60 (±2.44)	50.23 (±2.26)	0.900	93.25 (±0.45)
UNIREX (DLM-FP)	0.897	0.756	0.319 (±0.090)	0.167 (±0.036)	1.000	85.80 (±0.74)	72.76 (±0.19)	0.935	93.81 (±0.54)
UNIREX (SLM-FP)	0.891	0.807	0.302 (±0.039)	0.113 (±0.013)	0.940	82.55 (±0.84)	70.65 (±0.44)	0.927	93.68 (±0.67)

Table 4: **Main Results on SST.**

Method	Composite	Faithfulness			Plausibility			Performance	
	NRG (↑)	NRG (↑)	Comp (↑)	Suff (↓)	NRG (↑)	AUPRC (↑)	TF1 (↑)	NRG (↑)	F1 (↑)
AA (Grad)	0.481	0.457	0.184 (±0.023)	0.107 (±0.017)	0.028	13.31 (±0.91)	5.02 (±0.00)	0.957	95.33 (±0.65)
AA (Input*Grad)	0.503	0.359	0.148 (±0.031)	0.137 (±0.019)	0.194	8.68 (±0.37)	37.58 (±0.55)	0.957	95.33 (±0.65)
AA (DeepLIFT)	0.468	0.259	0.122 (±0.029)	0.172 (±0.022)	0.187	9.00 (±0.16)	36.15 (±1.45)	0.957	95.33 (±0.65)
AA (IG)	0.439	0.173	0.134 (±0.016)	0.219 (±0.044)	0.188	8.88 (±0.21)	36.39 (±1.29)	0.957	95.33 (±0.65)
L2E	0.550	0.445	0.000 (±0.007)	0.026 (±0.015)	0.248	16.68 (±10.20)	38.92 (±4.07)	0.957	95.33 (±0.65)
SGT	0.553	0.474	0.124 (±0.053)	0.071 (±0.064)	0.184	10.05 (±1.23)	34.64 (±1.67)	1.000	96.33 (±0.76)
FRESH	0.645	0.732	0.234 (±0.034)	0.000 (±0.000)	0.305	17.02 (±6.22)	48.26 (±5.87)	0.899	94.00 (±1.44)
A2R	0.431	0.764	0.267 (±0.050)	0.000 (±0.000)	0.244	35.44 (±21.69)	19.78 (±25.56)	0.284	79.78 (±7.14)
UNIREX (AA-F)	0.601	0.744	0.505 (±0.134)	0.122 (±0.100)	0.189	9.14 (±2.51)	36.28 (±1.84)	0.870	93.33 (±1.61)
SGT+P	0.586	0.604	0.152 (±0.013)	0.022 (±0.004)	0.183	9.16 (±1.59)	35.33 (±0.41)	0.971	95.66 (±1.16)
FRESH+P	0.491	0.691	0.193 (±0.062)	0.000 (±0.000)	0.710	65.78 (±11.16)	68.70 (±15.78)	0.070	74.84 (±12.22)
A2R+P	0.585	0.764	0.267 (±0.076)	0.000 (±0.000)	0.991	93.53 (±0.93)	88.77 (±1.22)	0.000	73.22 (±0.75)
UNIREX (DLM-P)	0.667	0.024	0.024 (±0.003)	0.238 (±0.004)	1.000	94.32 (±0.12)	89.53 (±1.63)	0.978	95.83 (±0.29)
UNIREX (AA-FP)	0.543	0.514	0.428 (±0.174)	0.195 (±0.105)	0.193	8.53 (±0.46)	37.71 (±3.12)	0.921	94.50 (±1.00)
UNIREX (DLM-FP)	0.744	0.326	0.283 (±0.217)	0.216 (±0.005)	0.991	93.65 (±0.36)	88.68 (±2.29)	0.913	94.33 (±1.61)
UNIREX (SLM-FP)	0.754	0.362	0.313 (±0.059)	0.213 (±0.014)	0.965	91.70 (±1.84)	86.17 (±1.20)	0.935	94.83 (±0.76)

Table 5: **Main Results on Movies.**

Method	Composite	Faithfulness			Plausibility			Performance	
	NRG (↑)	NRG (↑)	Comp (↑)	Suff (↓)	NRG (↑)	AUPRC (↑)	TF1 (↑)	NRG (↑)	Acc (↑)
AA (Grad)	0.537	0.504	0.331 (±0.012)	0.352 (±0.007)	0.130	37.33 (±0.62)	22.65 (±0.00)	0.977	63.56 (±1.27)
AA (Input*Grad)	0.573	0.361	0.249 (±0.018)	0.385 (±0.008)	0.383	39.56 (±0.54)	44.43 (±0.40)	0.977	63.56 (±1.27)
AA (DeepLIFT)	0.605	0.346	0.254 (±0.035)	0.403 (±0.042)	0.491	42.82 (±1.83)	51.72 (±1.26)	0.977	63.56 (±1.27)
AA (IG)	0.578	0.327	0.216 (±0.007)	0.378 (±0.010)	0.429	40.07 (±5.47)	48.34 (±3.16)	0.977	63.56 (±1.27)
L2E	0.544	0.493	0.005 (±0.003)	0.010 (±0.008)	0.161	23.56 (±1.09)	37.80 (±1.10)	0.977	63.56 (±1.27)
SGT	0.618	0.367	0.197 (±0.040)	0.324 (±0.015)	0.491	43.68 (±4.68)	51.00 (±3.05)	0.995	64.35 (±0.46)
FRESH	0.302	0.546	0.037 (±0.036)	0.000 (±0.000)	0.261	32.35 (±7.66)	39.37 (±0.70)	0.101	24.81 (±3.46)
A2R	0.277	0.516	0.014 (±0.021)	0.000 (±0.000)	0.282	41.61 (±3.85)	33.12 (±9.06)	0.032	21.77 (±1.31)
UNIREX (AA-F)	0.690	0.538	0.297 (±0.141)	0.286 (±0.084)	0.554	46.97 (±3.41)	53.99 (±1.66)	0.978	63.58 (±0.61)
SGT+P	0.601	0.367	0.201 (±0.032)	0.328 (±0.022)	0.436	41.30 (±6.70)	47.95 (±1.65)	1.000	64.57 (±0.33)
FRESH+P	0.374	0.515	0.013 (±0.021)	0.013 (±0.021)	0.606	53.40 (±12.87)	53.17 (±7.83)	0.000	20.36 (±0.66)
A2R+P	0.488	0.500	0.001 (±0.001)	0.000 (±0.000)	0.951	73.59 (±0.81)	67.63 (±1.54)	0.012	20.91 (±0.48)
UNIREX (DLM-P)	0.751	0.267	0.180 (±0.016)	0.390 (±0.035)	0.997	76.07 (±1.63)	69.76 (±0.27)	0.990	64.13 (±0.46)
UNIREX (AA-FP)	0.685	0.551	0.395 (±0.109)	0.381 (±0.101)	0.537	45.21 (±4.46)	53.91 (±3.23)	0.968	63.14 (±0.33)
UNIREX (DLM-FP)	0.814	0.492	0.293 (±0.043)	0.321 (±0.070)	0.997	76.38 (±0.57)	69.52 (±0.24)	0.953	62.50 (±1.34)
UNIREX (SLM-FP)	0.807	0.494	0.390 (±0.087)	0.424 (±0.110)	0.983	75.12 (±0.41)	69.25 (±0.41)	0.944	62.09 (±2.12)

Table 6: **Main Results on CoS-E.**

Method	Composite	Faithfulness			Plausibility			Performance	
	NRG (↑)	NRG (↑)	Comp (↑)	Suff (↓)	NRG (↑)	AUPRC (↑)	TF1 (↑)	NRG (↑)	F1 (↑)
AA (Grad)	0.498	0.462	0.222 (±0.028)	0.120 (±0.018)	0.035	22.27 (±0.17)	13.81 (±0.00)	0.997	69.80 (±0.60)
AA (Input*Grad)	0.506	0.289	0.225 (±0.048)	0.260 (±0.059)	0.231	18.51 (±0.23)	43.45 (±0.05)	0.997	69.80 (±0.60)
AA (DeepLIFT)	0.493	0.249	0.225 (±0.012)	0.292 (±0.014)	0.234	18.80 (±0.19)	43.51 (±0.04)	0.997	69.80 (±0.60)
AA (IG)	0.499	0.280	0.162 (±0.086)	0.222 (±0.086)	0.220	18.71 (±0.40)	41.79 (±1.33)	0.997	69.80 (±0.60)
L2E	0.522	0.366	0.007 (±0.006)	0.042 (±0.024)	0.205	24.48 (±2.71)	32.63 (±6.12)	0.997	69.80 (±0.60)
SGT	0.594	0.564	0.214 (±0.105)	0.033 (±0.077)	0.224	18.60 (±0.42)	42.42 (±0.51)	0.995	69.73 (±0.13)
FRESH	0.675	0.571	0.176 (±0.029)	0.000 (±0.000)	0.617	24.68 (±7.98)	48.02 (±3.04)	0.838	64.47 (±3.41)
A2R	0.217	0.404	-0.010 (±0.029)	0.000 (±0.000)	0.249	18.72 (±0.67)	45.45 (±0.02)	0.000	36.39 (±0.00)
UNIREX (AA-F)	0.711	0.956	0.505 (±0.050)	-0.071 (±0.020)	0.236	18.82 (±0.40)	43.68 (±0.38)	0.939	66.17 (±4.58)
SGT+P	0.630	0.665	0.280 (±0.029)	0.283 (±0.039)	0.226	18.63 (±0.52)	42.71 (±0.39)	1.000	69.91 (±0.81)
FRESH+P	0.404	0.413	0.000 (±0.013)	0.000 (±0.000)	0.739	55.87 (±10.13)	63.70 (±9.58)	0.060	38.41 (±5.34)
A2R+P	0.516	0.422	0.011 (±0.024)	0.000 (±0.000)	0.977	70.86 (±1.30)	76.21 (±1.68)	0.150	41.42 (±8.73)
UNIREX (DLM-P)	0.708	0.123	0.127 (±0.010)	0.322 (±0.017)	0.999	71.80 (±0.27)	77.94 (±0.57)	1.000	69.91 (±0.76)
UNIREX (AA-FP)	0.706	1.000	0.545 (±0.045)	-0.077 (±0.099)	0.231	19.13 (±0.71)	42.66 (±1.18)	0.888	66.17 (±4.58)
UNIREX (DLM-FP)	0.751	0.327	0.135 (±0.072)	0.165 (±0.029)	0.998	71.89 (±0.41)	77.63 (±0.62)	0.929	67.53 (±1.06)
UNIREX (SLM-FP)	0.784	0.377	0.198 (±0.038)	0.171 (±0.027)	0.997	71.69 (±0.21)	77.79 (±0.09)	0.979	69.20 (±1.58)

Table 7: **Main Results on MultiRC.**

Method	Composite	Faithfulness			Plausibility			Performance	
	NRG (↑)	NRG (↑)	Comp (↑)	Suff (↓)	NRG (↑)	AUPRC (↑)	TF1 (↑)	NRG (↑)	F1 (↑)
AA (Grad)	0.587	0.518	0.313 (±0.009)	0.380 (±0.025)	0.244	59.80 (±1.32)	15.27 (±0.00)	0.999	90.78 (±0.27)
AA (Input*Grad)	0.503	0.287	0.205 (±0.005)	0.446 (±0.020)	0.223	32.98 (±1.37)	43.13 (±0.86)	0.999	90.78 (±0.27)
AA (DeepLIFT)	0.508	0.270	0.195 (±0.012)	0.448 (±0.014)	0.254	33.47 (±1.31)	46.44 (±0.04)	0.999	90.78 (±0.27)
AA (IG)	0.596	0.473	0.308 (±0.011)	0.414 (±0.020)	0.317	47.83 (±1.04)	37.87 (±1.39)	0.999	90.78 (±0.27)
L2E	0.606	0.460	0.009 (±0.015)	0.036 (±0.022)	0.358	58.11 (±0.97)	31.35 (±0.27)	0.999	90.78 (±0.27)
SGT	0.595	0.503	0.288 (±0.025)	0.361 (±0.038)	0.298	42.46 (±3.03)	41.70 (±1.78)	0.985	90.23 (±0.16)
FRESH	0.518	0.661	0.120 (±0.075)	0.000 (±0.000)	0.361	38.77 (±6.82)	53.71 (±3.30)	0.530	72.92 (±8.71)
A2R	0.273	0.564	0.053 (±0.048)	0.000 (±0.000)	0.256	48.48 (±11.14)	29.54 (±24.72)	0.000	52.72 (±14.08)
UNIREX (AA-F)	0.622	0.539	0.330 (±0.018)	0.383 (±0.055)	0.340	45.29 (±3.02)	43.69 (±1.98)	0.987	90.31 (±0.19)
SGT+P	0.608	0.524	0.286 (±0.034)	0.339 (±0.032)	0.311	43.03 (±1.69)	42.59 (±1.63)	0.988	90.36 (±0.08)
FRESH+P	0.614	0.695	0.143 (±0.072)	0.000 (±0.000)	0.603	56.21 (±10.47)	64.09 (±5.59)	0.544	73.44 (±12.88)
A2R+P	0.800	0.751	0.182 (±0.097)	0.000 (±0.000)	0.992	87.30 (±0.44)	77.31 (±0.72)	0.656	77.31 (±0.72)
UNIREX (DLM-P)	0.842	0.525	0.311 (±0.011)	0.371 (±0.032)	1.000	87.85 (±0.13)	77.63 (±0.35)	1.000	90.80 (±0.33)
UNIREX (AA-FP)	0.626	0.529	0.341 (±0.008)	0.406 (±0.046)	0.363	44.79 (±0.81)	47.18 (±0.83)	0.985	90.21 (±0.08)
UNIREX (DLM-FP)	0.857	0.588	0.335 (±0.018)	0.346 (±0.023)	0.991	86.99 (±0.40)	77.53 (±0.15)	0.992	90.51 (±0.12)
UNIREX (SLM-FP)	0.864	0.603	0.353 (±0.017)	0.356 (±0.015)	0.994	87.58 (±0.14)	77.22 (±0.28)	0.994	90.59 (±0.09)

Table 8: **Main Results on e-SNLI.**

Ablation	Method	Performance	Faithfulness			Plausibility	
		Acc (↑)	CSD (↑)	Comp (↑)	Suff (↓)	AUPRC (↑)	TF1 (↑)
Ext Type (F)	UNIREX (AA-F, Rand)	94.05 (±0.35)	-0.156 (±-0.156)	0.171 (±0.040)	0.327 (±0.050)	44.92 (±0.00)	46.15 (±0.00)
	UNIREX (AA-F, Gold)	93.81 (±0.54)	-0.017 (±0.070)	0.232 (±0.088)	0.249 (±0.021)	100.00 (±0.00)	100.00 (±0.00)
	UNIREX (AA-F, Inv)	93.47 (±1.81)	-0.115 (±0.018)	0.242 (±0.010)	0.357 (±0.019)	20.49 (±0.00)	0.00 (±0.00)
	UNIREX (AA-F, IG)	93.81 (±0.55)	-0.138 (±0.040)	0.119 (±0.009)	0.258 (±0.031)	49.94 (±1.77)	50.75 (±0.54)
Ext Type (FP)	UNIREX (AA-FP, Sum)	93.81 (±0.55)	-0.138 (±0.040)	0.119 (±0.009)	0.258 (±0.031)	49.94 (±1.77)	50.75 (±0.54)
	UNIREX (AA-FP, MLP)	93.23 (±0.92)	0.087 (±0.134)	0.285 (±0.051)	0.197 (±0.100)	54.82 (±1.97)	49.62 (±0.65)
	UNIREX (DLM-FP)	93.81 (±0.18)	0.151 (±0.056)	0.319 (±0.090)	0.167 (±0.036)	85.80 (±0.74)	72.76 (±0.19)
	UNIREX (SLM-FP)	93.68 (±0.67)	0.189 (±0.030)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	70.65 (±0.44)
Comp/Suff Loss	UNIREX (SLM-FP, Comp)	93.59 (±0.11)	0.040 (±0.096)	0.350 (±0.048)	0.310 (±0.049)	82.79 (±0.62)	70.74 (±0.81)
	UNIREX (SLM-FP, Suff)	94.16 (±0.39)	0.014 (±0.010)	0.166 (±0.003)	0.152 (±0.012)	83.74 (±0.84)	70.94 (±0.86)
	UNIREX (SLM-FP, Comp+Suff)	93.68 (±0.67)	0.189 (±0.030)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	70.65 (±0.44)
Suff Criterion	UNIREX (SLM-FP, KL Div)	93.06 (±0.25)	0.174 (±0.100)	0.306 (±0.098)	0.131 (±0.005)	82.62 (±0.88)	70.43 (±0.65)
	UNIREX (SLM-FP, MAE)	93.78 (±0.13)	0.135 (±0.053)	0.278 (±0.058)	0.143 (±0.008)	82.66 (±0.61)	70.25 (±0.45)
	UNIREX (SLM-FP, Margin)	93.68 (±0.67)	0.189 (±0.030)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	70.65 (±0.44)
SLM Ext Head	UNIREX (SLM-FP, Linear)	93.68 (±0.67)	0.189 (±0.030)	0.302 (±0.039)	0.113 (±0.013)	82.55 (±0.84)	70.65 (±0.44)
	UNIREX (SLM-FP, MLP-2048-2)	93.67 (±0.18)	0.179 (±0.060)	0.323 (±0.071)	0.144 (±0.012)	83.82 (±0.77)	70.93 (±0.87)
	UNIREX (SLM-FP, MLP-4096-3)	93.19 (±0.79)	0.141 (±0.030)	0.295 (±0.057)	0.154 (±0.027)	84.53 (±0.61)	71.41 (±0.91)

Table 9: **UNIREX Ablation Studies on SST.**

Task	Dataset	Method	Performance	Faithfulness		
			Perf (↑)	CSD (↑)	Comp (↑)	Suff (↓)
Sentiment Analysis	SST	Vanilla	93.81 (±0.74)	-0.070 (±0.061)	0.145 (±0.023)	0.215 (±0.038)
		UNIREX (AA-F)	93.19 (±0.40)	0.360 (±0.055)	0.405 (±0.031)	0.045 (±0.024)
		UNIREX (DLM-FP)	93.81 (±0.18)	0.151 (±0.056)	0.319 (±0.090)	0.167 (±0.036)
		UNIREX (SLM-FP)	93.68 (±0.67)	0.189 (±0.030)	0.302 (±0.039)	0.113 (±0.013)
	Yelp	Vanilla	92.50 (±2.07)	-0.156 (±0.028)	0.067 (±0.004)	0.222 (±0.031)
		UNIREX (AA-F)	90.75 (±1.30)	-0.138 (±0.120)	0.096 (±0.026)	0.233 (±0.096)
		UNIREX (DLM-FP)	92.37 (±0.46)	0.169 (±0.060)	0.265 (±0.094)	0.097 (±0.033)
		UNIREX (SLM-FP)	86.60 (±1.57)	0.114 (±0.056)	0.175 (±0.055)	0.060 (±0.001)
	Amazon	Vanilla	91.13 (±0.28)	-0.120 (±0.038)	0.096 (±0.008)	0.217 (±0.033)
		UNIREX (AA-F)	86.60 (±0.95)	-0.111 (±0.161)	0.100 (±0.042)	0.210 (±0.122)
		UNIREX (DLM-FP)	89.35 (±2.22)	0.133 (±0.039)	0.232 (±0.072)	0.098 (±0.033)
		UNIREX (SLM-FP)	81.82 (±7.62)	0.097 (±0.027)	0.147 (±0.012)	0.050 (±0.017)
Hate Speech Detection	Stormfront	Vanilla	10.48 (±1.66)	-0.066 (±0.072)	0.153 (±0.002)	0.219 (±0.071)
		UNIREX (AA-F)	9.43 (±1.45)	0.329 (±0.104)	0.337 (±0.073)	0.008 (±0.031)
		UNIREX (DLM-FP)	10.37 (±2.66)	0.052 (±0.027)	0.167 (±0.084)	0.115 (±0.059)
		UNIREX (SLM-FP)	4.51 (±1.87)	0.049 (±0.041)	0.110 (±0.039)	0.062 (±0.043)
Offensive Speech Detection	OffenseEval	Vanilla	33.51 (±0.99)	-0.125 (±0.068)	0.104 (±0.007)	0.229 (±0.064)
		UNIREX (AA-F)	35.69 (±2.30)	-0.028 (±0.084)	0.076 (±0.008)	0.104 (±0.076)
		UNIREX (DLM-FP)	35.52 (±1.26)	0.053 (±0.012)	0.140 (±0.049)	0.087 (±0.045)
		UNIREX (SLM-FP)	38.17 (±0.96)	0.039 (±0.031)	0.087 (±0.016)	0.048 (±0.024)
Irony Detection	SemEval2018-Irony	Vanilla	29.63 (±4.72)	-0.058 (±0.075)	0.154 (±0.001)	0.212 (±0.074)
		UNIREX (AA-F)	47.99 (±6.33)	0.026 (±0.080)	0.087 (±0.022)	0.061 (±0.071)
		UNIREX (DLM-FP)	31.97 (±2.80)	0.047 (±0.017)	0.149 (±0.052)	0.102 (±0.053)
		UNIREX (SLM-FP)	17.42 (±4.04)	0.027 (±0.047)	0.091 (±0.027)	0.064 (±0.033)

Table 10: **Zero-Shot Explainability Transfer from SST to Unseen Datasets/Tasks.**