Revisiting Pathologies of Neural Models under Input Reduction

Canasai Kruengkrai **Junichi Yamagishi**
National Institute of Informatics, Japan
{canasai,jyamagishi}@nii.ac.jp

Abstract

We revisit the question of why neural models tend to produce high-confidence predictions on inputs that appear nonsensical to humans. Previous work has suggested that the models fail to assign low probabilities to such inputs due to model overconfidence. We evaluate various regularization methods on fact verification benchmarks and find that this problem persists even with well-calibrated or underconfident models, suggesting that overconfidence is not the only underlying cause. We also find that regularizing the models with reduced examples helps improve interpretability but comes with the cost of miscalibration. We show that although these reduced examples are incomprehensible to humans, they can contain valid statistical patterns in the dataset utilized by the model.[1]

Evidence Dataset: COVIDFACT

CONCLUSIONS : In our cohort of COVID-19 patients, immunosuppression was associated with a lower risk of moderate-severe ARDS.

Original supported claim

Immunosuppression is associated with a **lower** risk of moderate to severe acute respiratory distress syndrome in covid-19 .

Reduced supported claim

upp moderate respiratory .

Confidence 1.000 → 0.999

Original refuted claim

Immunosuppression is associated with a **higher** risk of moderate to severe acute respiratory distress syndrome in covid-19.

Reduced refuted claim

is associated

Confidence 0.999 → 0.904

Figure 1: Examples of the original and reduced claims from the COVIDFACT test set where the model still makes the same correct predictions without considering the salient words (highlighted in blue and red). These reduced claims are ungrammatical/uninformative and appear random to humans.

1 Introduction

During the development stage, we put much effort into tuning neural models to achieve high accuracy on held-out data. However, when deploying such tuned models in real-world scenarios, it is also important for them to be reliable. For example, when a fact verification model judges that a claim is true with a confidence of 0.95, it should have a 95% chance of being correct. Meanwhile, low-confidence predictions can be passed onto humans to be double-checked manually. If the model can align its confidence with the correctness, it is considered calibrated. Despite achieving human-level performance on various tasks, recent studies (Guo et al., 2017; Ovadia et al., 2019; Hendrycks et al., 2020) have shown that modern neural models tend to be miscalibrated.

Miscalibration further reveals an anomaly of neural models in which they tend to produce high-confidence predictions on inputs that appear nonsensical to humans. Figure 1 shows examples from the COVIDFACT dataset (Saakyan et al., 2021) where the fact verification model still makes the same correct prediction given the reduced version of the original claim. Feng et al. (2018) first discovered such pathologies of neural models on widely used NLP datasets, such as SQUAD (Rajpurkar et al., 2016) and SNLI (Bowman et al., 2015). They attributed the main underlying cause to model overconfidence and proposed a regularization method incorporating reduced examples to mitigate the problem. While the interpretability could be improved, it is unclear how the reduced examples affect model calibration. In addition, their method is based on an entropy regularizer called the confidence penalty (Pereyra et al., 2017), and other possible techniques still remain uninvestigated.

In this paper, we explore a family of regularization methods and propose an extension that unifies label smoothing (Szegedy et al., 2016) and

[1]Our code is available at https://github.com/nii-yamagishilab/pathologies.

Findings of the Association for Computational Linguistics: ACL 2023, pages 11504–11517
July 9-14, 2023 ©2023 Association for Computational Linguistics

the confidence penalty (Pereyra et al., 2017). We conducted experiments on three fact verification datasets and found that:

- Pathologies still occur even when the model is well-calibrated or underconfident.
- Incorporating the reduced examples improves interpretability (i.e., increases the input lengths) but amplifies miscalibration (i.e., increases calibration errors).

Our results suggest that model overconfidence is not the only cause of pathological behaviors. Regularizing the objective function with the reduced examples encourages the model to output high entropy (i.e., low confidence) on such examples. However, these reduced examples can also contain valid statistical patterns that are sufficient for the model (but nonsensical to humans) to make predictions. This finding has also been observed in computer vision (Carter et al., 2021).

2 Task formulation

2.1 Datasets

We focus on the task of fact verification, which involves classifying a claim as supported (SUP), refuted (REF), or not enough information (NEI) with respect to evidence. We conduct experiments on three datasets:

COVIDFACT (Saakyan et al., 2021) starts from valid real-world claims and evidence sentences from peer-reviewed research documents concerning the COVID-19 pandemic. They then generated counterclaims by replacing the most salient word in the original claim using language model infilling with entailment-based quality control. The dataset consists of 3,263/419/404 samples in the training/dev/test sets with two classes: SUP and REF.

FEVER (Thorne et al., 2018) is from the Fact Extraction and VERification challenge, which has three subtasks: document retrieval, sentence selection, and fact verification. We only consider fact verification and use the data preprocessed by Schuster et al. (2021), which consists of 178,059/11,620/11,710 samples in the training/dev/test sets with three classes: SUP, REF, and NEI.

VITAMINC (Schuster et al., 2021) augments FEVER with the symmetric annotation strategy (Schuster et al., 2019). Given a claim-evidence pair from FEVER, they first edited the evidence sentence to flip the original label (e.g., REF→SUP) and then composed a new claim that holds the original label for the new, edited evidence sentence. They also collected new samples from Wikipedia revisions, but we only use the synthetically created dataset, which consists of 121,700/20,764/20,716 samples in the training/dev/test sets with two classes: SUP and REF.

2.2 Architecture

We formulate our task as supervised multi-class classification. Our aim is to train a model that can assign a label $y \in \mathcal{Y} = \{1, \ldots, K\}$ to an input $x \in \mathcal{X}$. Our model is a neural network h parameterized by $\boldsymbol{\theta}$:

$$h_{\boldsymbol{\theta}}(x) = \text{MLP}(\text{PLM}(x)),$$

where MLP is a multilayer perceptron and PLM is a pre-trained language model. Each PLM layer transforms x into a sequence of hidden state vectors.[2] Following standard practice, we obtain the fixed-length vector representation of x from the first hidden state vector of the last PLM layer. The MLP then maps the vector representation to K unnormalized logits. Finally, we apply the softmax function to obtain the predicted distribution $p \in \mathbb{R}^K$ over labels:

$$p(y|x) = \text{softmax}(h_{\boldsymbol{\theta}}(x)).$$

Let $q \in \mathbb{R}^K$ denote the ground-truth label distribution (i.e., one-hot encoding). During training, we aim to minimize the cross-entropy loss between q and p:

$$L_{ce} = \text{H}(q, p) = \sum_{y \in \mathcal{Y}} q(y|x) \log \frac{1}{p(y|x)}. \quad (1)$$

3 Input Reduction

Model interpretation methods offer explanations for model predictions (Ribeiro et al., 2016; Li et al., 2016; Wallace et al., 2019). The goal is to understand why the model made specific predictions. A brute-force method is to look at model weights, but they are incomprehensible. Because most modern neural architectures (including ours) rely on attention mechanisms, attention weights over inputs are often used as explanations. However, subsequent

[2] In our case, an input x is a concatenation of claim and evidence sentences.

Algorithm 1 Input reduction

Require: Original x
$\hat{y} = \text{argmax}_{y \in \mathcal{Y}}\, p(y|x)$
while true **do**
  $w^* = \text{argmin}_{w \in x} \|\nabla_{\mathbf{e}_w} L_{ce}\|$
  $\tilde{x} \leftarrow x \setminus w^*$
  $\tilde{y} = \text{argmax}_{y \in \mathcal{Y}}\, p(y|\tilde{x})$
  if $\tilde{y} == \hat{y}$ and $\tilde{x} \neq \varnothing$ **then**
    $x \leftarrow \tilde{x}$
  else
    break
  end if
end while
return Final x

Evidence Dataset: COVIDFACT

Toms Hardware reports that The Raspberry Pi Foundation is ramping up production of its Pi Zero boards to help supply manufacturers with enough units to keep up with the high demand for ventilators. ... (*truncated*)

Original refuted claim

Raspberry pi about to **avoid** ventilators for coronavirus victims

Reduced refuted claim

(0.999) R aspberry pi about to **avoid** vent il ators for coron av irus victims
(0.999) aspberry pi about to **avoid** vent il ators for coron av irus victims
(0.999) pi about to **avoid** vent il ators for coron av irus victims
(0.999) pi about to **avoid** vent ators for coron av irus victims
(0.997) pi about to **avoid** vent ators for av irus victims
(0.995) about to **avoid** vent ators for av irus victims
(0.997) to **avoid** vent ators for av irus victims
(0.989) **avoid** vent ators for av irus victims
(0.986) vent ators for av irus victims
(0.988) ators for av irus victims
(0.989) ators for av irus

Figure 2: Reduction path of the refuted claim from the COVIDFACT dev set. The number in parentheses indicates model confidence, which remains high during reduction. Although the salient word "avoid" is removed, the model still makes the same correct prediction with 0.989 confidence.

studies have argued that attention weights can be manipulated (Pruthi et al., 2020) and uncorrelated with feature importance measures (Jain and Wallace, 2019).

In our work, we focus on a gradient-based method called input reduction (Feng et al., 2018). The idea is to find a minimal input subset sufficient for attaining the same prediction as the original input. This minimal input subset can be regarded as a *rationale*, i.e., a few substrings that are sufficient for justifying predictions (Zaidan et al., 2007).

Input reduction iteratively removes the least important word from the original input until the model changes its prediction. In our case, the basic unit is a token, which can be a word or a subword. Let $w \in x$ denote a token in the input and $\mathbf{e}_w$ denote its embedding vector obtained from the PLM. Algorithm 1 summarizes the process of our input

Evidence Dataset: FEVER

Epistemology studies the nature of knowledge, justification, and the rationality of belief.

Original refuted claim

Epistemology has nothing to do with the study of the rationality of belief.

Reduced refuted claim

nothing do

Confidence 0.963 → 0.946

Evidence Dataset: VITAMINC

Shortly after Plato died , Aristotle left Athens and , at the request of Philip II of Macedon , tutored Alexander the Great beginning in 343 BC .

Original supported claim

Aristotle tutored Alexander the Great .

Reduced supported claim

otle tut

Confidence 0.998 → 0.991

Figure 3: Additional examples of the original and reduced claims from the FEVER and VITAMINC dev sets, where the prediction of the reduced claim is identical to that of the original claim.

reduction. Note that the ground-truth label is unnecessary for input reduction. We estimate the importance of each w through the hallucinated gradient of the loss with respect to the embedding vector and the predicted label. At each iteration, we remove the token having the smallest gradient norm (Wallace et al., 2019). We only proceed if the new predicted label of the reduced input $\tilde{x}$ is the same as that of the original input x.

Our inspection

Recall that our input is a sentence pair consisting of claim and evidence sentences. To conform with Feng et al. (2018), we remove tokens from the claim only (equivalent to the hypothesis in SNLI) and keep the evidence untouched. Figure 2 shows the reduction path of the refuted claim from the COVIDFACT dev set generated using Algorithm 1. Figure 3 shows additional examples of the original and reduced claims from FEVER and VITAMINC.

Figure 4 compares the claim lengths before and after reduction on the FEVER, VITAMINC, and COVIDFACT dev sets. Unlike Feng et al. (2018), we examine the results in detail by class. Feng et al. (2018) reported that the reduced examples only contain one or two words on average across all of their tasks. However, we find that their observation holds on particular classes on our specific datasets. The NEI/REF claims can be reduced to a few tokens without changing the original predic-

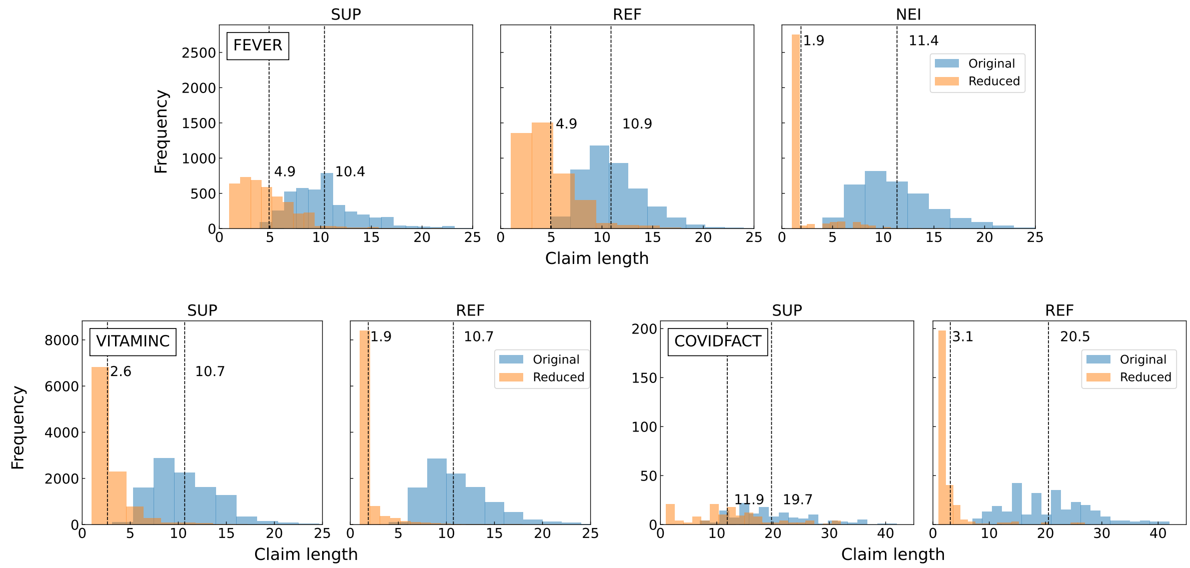

Figure 4: Distributions of claim lengths before and after input reduction on the dev sets of FEVER, VITAMINC, and COVIDFACT. The SUP claims cannot be reduced to very short lengths to retain the original predictions, in contrast to the REF/NEI claims.

tions. On the contrary, we observe that the SUP claims need to remain longer to retain the original predictions.

Our observation seems to correlate with fact-checking data construction. The process usually starts with creating valid claims (i.e., SUP) and modifying them to create other types (REF/NEI), which leaves annotation artifacts (Gururangan et al., 2018) or shortcuts (Geirhos et al., 2020), enabling the model to use them for predictions.

4 Regularization methods

In this section, we review widely used regularization methods, inspect their properties, and introduce our extension.

4.1 Existing methods

Temperature scaling (Guo et al., 2017) is a simple yet effective regularization method that simplifies Platt scaling (Platt, 1999) by adjusting the unnormalized logits with only one parameter, temperature $\tau \in \mathbb{R}$:

$$p(y|x) = \text{softmax}(\frac{h_{\boldsymbol{\theta}}(x)}{\tau}).$$

We can soften the predicted distribution by setting $\tau > 1$. Following Guo et al. (2017), we use temperature scaling as a post-processing method so that the model accuracy is preserved (i.e., the predicted labels remain unchanged). We optimize τ with respect to L_{ce} (defined in Eq. (1)) on the development set. This procedure differs from the softmax temperature used in knowledge distillation (Hinton et al., 2015), which involves training a small model with the soft target labels from a larger model.

Label smoothing (Szegedy et al., 2016), in contrast to temperature scaling, softens the ground-truth label distribution q. Label smoothing replaces q with $q' = (1-\epsilon)q + \epsilon u(y)$, where ϵ is a balancing parameter, and $u(y)$ is the uniform distribution over labels (i.e., $u(y) = \frac{1}{K}$). For notational convenience, we scale down q' by $1/(1-\epsilon)$ so that:

$$q'_s = q + \beta u(y),$$

where $\beta = \frac{\epsilon}{(1-\epsilon)}$ (Meister et al., 2020). By applying Eq. (1), we can derive the label smoothing loss as:

$$\begin{aligned}
L_{ls} &= \mathrm{H}(q'_s, p) \\
&= \sum_{y \in \mathcal{Y}} (q(y|x) + \beta u(y)) \log \frac{1}{p(y|x)} \\
&= \sum_{y \in \mathcal{Y}} q(y|x) \log \frac{1}{p(y|x)} \\
&\quad + \beta \sum_{y \in \mathcal{Y}} u(y) \log \frac{1}{p(y|x)} \\
&= L_{ce} + \beta\, \mathrm{H}(u, p). \qquad (2)
\end{aligned}$$

The above equation consists of the usual cross-entropy loss and the regularization function

H(u, p). It is also equivalent to the cross-entropy form of Szegedy et al.'s (2016) label smoothing.

Confidence penalty (Pereyra et al., 2017), as its name suggests, penalizes the confident predicted distribution p. We can measure the degree of confidence in p by using the entropy H(p). A high confidence p corresponds to a low H(p) and vice versa. Pereyra et al. (2017) defined the confidence penalty loss as:

$$L_{cp} = L_{ce} - \beta\,\mathrm{H}(p). \tag{3}$$

The regularization function of the above equation becomes the negative entropy H(p). The balancing parameter β enables a trade-off between minimizing the cross-entropy loss and maximizing the entropy of the predicted distribution p.

4.2 Observations

Guo et al. (2017) empirically found that model miscalibration is due to negative log-likelihood overfitting. Here, we interpret this phenomenon from a Kullback–Leibler (KL) divergence perspective. Let H(q) denote the entropy of the ground-truth label (one-hot) distribution, which is a constant. We rewrite the cross-entropy loss in Eq. (1) as:

$$\begin{aligned} L_{ce} &= \mathrm{H}(q,p) - \mathrm{H}(q) + \mathrm{H}(q) \\ &= \mathrm{KL}(q \parallel p) + \underbrace{\mathrm{H}(q)}_{\text{constant}} . \end{aligned} \tag{4}$$

Thus, minimizing L_{ce} is equivalent to minimizing the KL divergence between the ground-truth label distribution q and the predicted distribution p (i.e., pushing p towards q). When overfitting occurs, the model places most of the probability mass to a single label, resulting in peakiness in p. Typically, mitigating model miscalibration involves making p less peaky.

We can also express the label smoothing loss in KL divergence form. We know that:

$$\mathrm{KL}(u \parallel p) = \mathrm{H}(u,p) - \mathrm{H}(u), \tag{5}$$

Therefore, we can rewrite Eq. (2) as:

$$L_{ls} = L_{ce} + \beta\,\mathrm{KL}(u \parallel p) + \underbrace{\beta\,\mathrm{H}(u)}_{\text{constant}} .$$

Thus, minimizing L_{ls} is equivalent to finding a balance between pushing p towards q (as defined in Eq. (4)) and towards u. Likewise, we can express the confidence penalty loss in (reverse) KL divergence form. Since:

$$\mathrm{KL}(p \parallel u) = \mathrm{H}(p,u) - \mathrm{H}(p), \tag{6}$$

we reformulate Eq. (3) as:

$$L_{cp} = L_{ce} + \beta\,\mathrm{KL}(p \parallel u) - \underbrace{\beta\,\mathrm{H}(p,u)}_{\text{constant}} .$$

Since the KL divergence is always non-negative, it follows from Eqs. (5) and (6) that H(p) is upper bounded by H(u, p):

$$\mathrm{H}(u,p) \geq \mathrm{H}(u) = \mathrm{H}(p,u)^4 \geq \mathrm{H}(p).$$

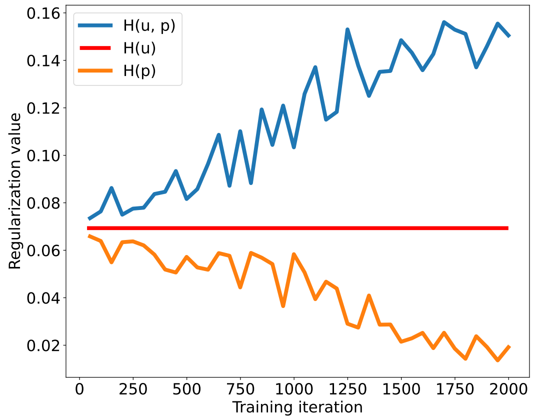

Figure 5: Regularization terms H(u, p) and H(p) in L_{ls} and L_{cp}, respectively, trained on COVIDFACT. We also include H(u) as a reference value.[3] H(u, p) and H(p) start close to H(u) and then diverge as the models become more confident in their predictions, resulting in low H(p) but high H(u, p).

We inspect the above relationship by plotting H(u, p) and H(p) in L_{ls} and L_{cp}, respectively, as shown in Figure 5. We trained the models for 10 epochs with $\beta = 0.1$. Each epoch can have many iterations depending on the mini-batch size.

Interestingly, both curves appear to be mirror images of each other in the early iterations. H(u, p) and H(p) start close to H(u), meaning that the models place almost equal probabilities on both labels. As the number of iterations increases, the models become more and more confident in their predictions, and H(u, p) and H(p) gradually diverge from H(u). Another observation is that H(p) heavily penalizes the confidence penalty loss in Eq. (3) at

[3] The number of classes in COVIDFACT is 2, so $\beta\,\mathrm{H}(u) = 0.1 \log(2) \approx 0.069$.

[4] The equation H(u) = H(p, u) follows from the fact that $\mathrm{H}(u) = \sum_{y \in \mathcal{Y}} u(y) \log \frac{1}{u(y)} = \log K$ and $\mathrm{H}(p,u) = \sum_{y \in \mathcal{Y}} p(y|x) \log \frac{1}{u(y)} = \log K$.

the beginning iterations because $\mathrm{H}(p)$ starts close to $\mathrm{H}(u)$ (i.e., the maximum entropy). However, the effect of $\mathrm{H}(p)$ diminishes because its value approaches zero at the final iterations. This behavior is contrary to that of $\mathrm{H}(u, p)$.

4.3 Proposed extension

Being able to represent L_{ls} and L_{cp} in asymmetric KL divergence forms encourages us to pursue their symmetric counterpart. A known symmetric form of the KL divergence is the Jeffreys (J) divergence (Jeffreys, 1946), defined as $\mathrm{J}(p_1 \parallel p_2) = \mathrm{KL}(p_1 \parallel p_2) + \mathrm{KL}(p_2 \parallel p_1)$.[5] On the basis of the J divergence, we derive our loss as:

$$\begin{aligned} L_J &= L_{ce} + \beta\, \mathrm{J}(u \parallel p) \\ &= L_{ce} + \beta \left(\mathrm{KL}(u \parallel p) + \mathrm{KL}(p \parallel u)\right) \\ &= L_{ce} + \beta \left(\mathrm{H}(u, p) - \mathrm{H}(p)\right). \end{aligned} \tag{7}$$

The regularization term of Eq. (7) simply becomes the combination of those of L_{ls} and L_{cp} from Eqs. (2) and (3), respectively.

5 Hybrid methods

Feng et al. (2018) proposed a regularization method to mitigate overconfident predictions on nonsensical inputs, specifically by modifying Pereyra et al.'s (2017) confidence penalty with the reduced examples. The idea resembles data augmentation, but they only used the reduced examples for computing the regularization function. They first applied input reduction (described in §3) to the original training set to obtain its reduced version $\widetilde{\mathcal{X}}$. Let $\tilde{p}(y|\tilde{x})$ denote the predicted distribution given the reduced example $\tilde{x} \in \widetilde{\mathcal{X}}$. By modifying Eq. (3), Feng et al.'s (2018) loss function can be expressed as:[6]

$$L_{\widetilde{cp}} = L_{ce} - \beta\, \mathrm{H}(\tilde{p}). \tag{8}$$

Therefore, the model will attempt to maximize $\mathrm{H}(\tilde{p})$ (i.e., making $\tilde{p}$ less peaky) to reduce the overall loss.

Proposed extension

Because the modification in Eq. (8) only involves the regularization function, this motivates us to apply the same idea to L_{ls} and L_J. From Eqs. (2) and (7), we derive two additional loss functions that incorporate the reduced examples:

$$L_{\widetilde{ls}} = L_{ce} + \beta\, \mathrm{H}(u, \tilde{p}), \tag{9}$$

and

$$L_{\widetilde{J}} = L_{ce} + \beta\, \mathrm{J}(u \parallel \tilde{p}). \tag{10}$$

6 Experiments

6.1 Training details

We implemented our model (described in §2.2) on top of Hugging Face's Transformers library (Wolf et al., 2020). For the PLM, we used RoBERTa-base (Liu et al., 2019). For optimization, we used Adafactor (Shazeer and Stern, 2018) with a learning rate of 3e-5, a linear learning rate decay, a warmup ratio of 0.02, and a gradient clipping of 1.0. We trained each model for 10 epochs or until the validation accuracy had not improved after three times (i.e., early stopping with a patience of 3). Early stopping can also be regarded as a regularization method to alleviate overfitting.

We used a batch size of 256 for FEVER and VITAMINC. Following Saakyan et al. (2021), we used a batch size of 16 for COVIDFACT. We found that using a large batch size yields lower accuracy on COVIDFACT. One plausible explanation is that COVIDFACT has a much smaller training set than FEVER and VITAMINC. We fixed the model hyperparameters and searched for an optimal β in the range of {0.05, 0.1, 0.3, 0.5} for the regularization methods (§4) and their variants (§5) on the dev set. We conducted all experiments on NVIDIA Tesla A100 GPUs.

6.2 Assessing model miscalibration

The common practice of assessing model miscalibration is to visualize the probability outputs with confidence histograms and reliability diagrams (Niculescu-Mizil and Caruana, 2005; Guo et al., 2017). Further, these visualizations can be summarized by a single number using the expected calibration error (Naeini et al., 2015).

Confidence histograms: Let $\hat{p}_j$ denote the confidence score of the j^{th} sample where $\hat{p}_j = \max_{y_j \in \mathcal{Y}} p(y_j|x_j)$. We first divide the confidence range of $[0, 1]$ into M equal-size bins. The i^{th} bin covers the interval of $(\frac{i-1}{M}, \frac{i}{M}]$. We then assign each $\hat{p}_j$ to its corresponding interval. To plot a confidence histogram, we compute the percentage of samples in each bin.

[5] Another symmetric form is the Jensen–Shannon (JS) divergence (Lin, 1991). We discuss its properties in Appendix A.

[6] Feng et al. (2018) formulated their problem as maximization, so the sign of the regularization term in their paper is positive.

Model	CovidFact				Fever				VitaminC			
	β	Acc	ECE	Len	β	Acc	ECE	Len	β	Acc	ECE	Len
L_{ce}	-	82.7	15.2	5.8	-	96.2	2.4	4.1	-	94.2	4.0	2.4
L_{ce+ts}	–	82.7	**14.0**	–	–	96.2	**2.0**	–	–	94.2	**3.5**	–
L_{ls}	0.10	84.7	9.8	5.2	0.05	96.2	1.8	3.7	0.05	94.1	1.9	2.4
L_{cp}	0.05	82.9	7.3	4.7	0.10	96.2	**1.5**	3.7	0.30	94.0	2.6	2.3
L_J	0.05	84.2	**6.6**	5.2	0.05	96.2	2.0	3.5	0.05	94.0	**1.7**	2.3
$L_{\widetilde{ls}}$	0.50	82.2	**7.4**	6.1	0.10	96.3	**1.9**	6.5	0.05	94.0	4.2	3.9
$L_{\widetilde{cp}}$	0.05	82.2	13.5	6.2	0.10	96.0	2.1	6.8	0.05	94.2	**4.1**	4.2
$L_{\widetilde{J}}$	0.50	83.7	10.6	7.2	0.10	96.2	2.1	7.0	0.05	94.0	**4.1**	4.3

Table 1: Results on CovidFact, Fever, and VitaminC test sets. We show the optimal β values found on the dev sets. Acc = accuracy; ECE = expected calibration error (lower is better); Len = average length of the claim after input reduction; L_{ce+ts} = L_{ce} post-processed with temperature scaling. The lowest ECE in each group is in bold.

Reliability diagrams: Let $\hat{y}_j$ denote the predicted label of the j^{th} sample where $\hat{y}_j = \text{argmax}_{y_j \in \mathcal{Y}}\, p(y_j|x_j)$ and $\mathcal{B}_i$ denote the set of samples belonging to the i^{th} bin. To plot a reliability diagram, we compute the average accuracy of the i^{th} bin:

$$\text{acc}(\mathcal{B}_i) = \frac{1}{|\mathcal{B}_i|} \sum_{j \in \mathcal{B}_i} \mathbb{1}(\hat{y}_j = y_j),$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Expected calibration error: In the same manner as $\text{acc}(\mathcal{B}_i)$, we compute the average confidence of the i^{th} bin:

$$\text{conf}(\mathcal{B}_i) = \frac{1}{|\mathcal{B}_i|} \sum_{j \in \mathcal{B}_i} \hat{p}_j.$$

The expected calibration error (ECE) is the weighted average of the gaps between $\text{acc}(\mathcal{B}_i)$ and $\text{conf}(\mathcal{B}_i)$ of all bins:

$$\text{ECE} = \sum_{i=1}^{M} \frac{|\mathcal{B}_i|}{N} |\text{acc}(\mathcal{B}_i) - \text{conf}(\mathcal{B}_i)|,$$

where N is the number of all samples.

6.3 Results

We report the accuracy (Acc), ECE, and average claim length (Len) after input reduction. The average length acts as a proxy for quick assessment of whether there are any differences among model's predictions. An increase in the length would mean that the reduced claims are less likely to appear nonsensical to humans (Feng et al., 2018), though further inspection would be necessary.

Effect of regularization

Our proposed L_J produces the lowest ECE on CovidFact and VitaminC, as shown in Table 1 (middle section). Generally, all entropy regularization models yield lower ECE than temperature scaling. Figure 6 compares the confidence histograms and reliability diagrams of the baseline model with those of the best regularization models. The baseline L_{ce} shows severe miscalibration on CovidFact. Our proposed L_J helps bridge the gaps between the accuracy and confidence of all bins. Surprisingly, L_{ce} already produces low ECE on Fever and VitaminC, while L_{cp} and L_J further improve the accuracy-confidence alignment.

The results on Fever and VitaminC also demonstrate that the models become underconfident in the last bin (i.e., the interval of $(0.95, 1]$), which contains most of the model's predictions. Feng et al. (2018) suggested that the pathological behaviors of the models is a consequence of model overconfidence. In contrast, our results show that this problem still occurs even when the model is well-calibrated or underconfident.

Effect of incorporating reduced examples in training

Table 1 (bottom section) shows the results of the hybrid models (described in §5), which augment the training set with the reduced examples and use them in the regularization function. During training, incorporating the reduced examples encourages the model to output high entropy (i.e., low confidence) on such examples. Consequently, during testing, the hybrid models can no longer reduce the input sentence to a very short length while maintaining high confidence. While these models

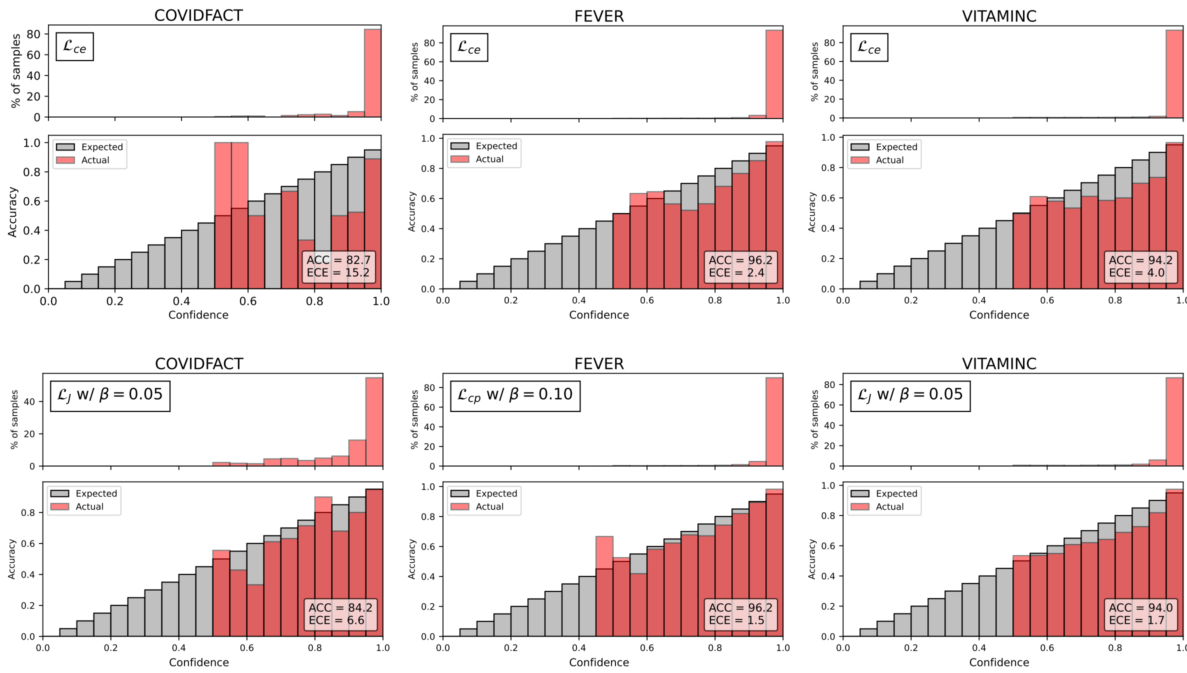

Figure 6: Confidence histograms and reliability diagrams ($M = 20$) for the baseline model (top) and the best regularization methods without data augmentation (bottom). On FEVER and VITAMINC, the baseline L_{ce} produces low ECE, and all models are slightly underconfident (i.e., more accurate than expected) in the last bin, which contains the majority of samples.

Dataset	Trained on	Evaluated on	Acc	ECE
COVIDFACT	Original	Original	82.7	15.2
	Original	Reduced	82.7	10.5
	Original	Random	56.9	34.5
	Reduced	Reduced	81.4	17.9
	Random	Random	74.8	23.0
FEVER	Original	Original	96.2	2.4
	Original	Reduced	96.2	6.3
	Original	Random	64.3	27.8
	Reduced	Reduced	94.1	3.3
	Random	Random	79.4	12.5
VITAMINC	Original	Original	94.2	4.0
	Original	Reduced	94.1	8.4
	Original	Random	62.4	23.4
	Reduced	Reduced	90.7	6.3
	Random	Random	72.2	7.7

Table 2: Results of training/evaluating on same/different datasets using our baseline L_{ce}. Original = original dataset; Reduced = dataset derived from applying input reduction on the original dataset and assigning the ground-truth labels; Random = dataset where each claim consists of tokens randomly sampled with the same length as the reduced claim.

increase the average length, they deteriorate ECE compared to their normal versions.

Are reduced examples valid statistical patterns in the dataset?

Following Carter et al. (2021), we constructed additional datasets from the reduced examples. Recall that input reduction relies on the predicted label from the model when producing reduced examples. The reduced example only maintains the original model prediction, which can be correct or incorrect. Here, we replaced the predicted label with the corresponding ground-truth label for each reduced example to create the reduced datasets. Thus, the reduced example is not the optimal representative of the original one with the true label. We can expect discrepancies to a certain extent.

Table 2 shows the results of our baseline L_{ce} on various settings. The original-original rows are from Table 1. We observe slight drops in accuracy when training/evaluating on the reduced datasets (i.e., reduced-reduced rows). The reduced examples produced by input reduction yield higher accuracy than those created by randomly selecting tokens in all settings. These results indicate that although the reduced examples do not align with human intuitions, they indeed contain valid statistical patterns in the datasets.

Model	Correct	w/ Salient	Success (%)
$\mathcal{L}_{ce}$	333	165	49.5
$\mathcal{L}_{ls}$	341	139	40.8
$\mathcal{L}_{cp}$	334	127	38.0
$\mathcal{L}_{J}$	339	150	44.2
$\mathcal{L}_{\widetilde{ls}}$	331	148	44.7
$\mathcal{L}_{\widetilde{cp}}$	331	199	**60.1**
$\mathcal{L}_{\widetilde{J}}$	337	123	36.5

Table 3: Results of capturing salient words on COVIDFACT test set. The "correct" column is the number of correct predictions, while the "w/ salient" column is the number of those that contain the salient word in the reduced claim.

Do longer reduced examples capture more meaningful information?

An ideal way to check whether longer reduced examples capture more meaningful information is to ask humans to evaluate the reduced claims, but this is time-consuming and costly. Here, we exploited a characteristic of COVIDFACT in which the counterclaim differs from the original claim in only one salient word, as shown in Figure 1. This enables us to perform the automatic evaluation. We first chose all reduced claims where the predictions are correct. We then checked whether the salient word in the original claim is present in the reduced claim.

Table 3 shows that $L_{\widetilde{cp}}$ captures more salient words than other models on COVIDFACT. Appendix B provides additional examples where $L_{\widetilde{cp}}$ can successfully retain salient words. However, the ECE of $L_{\widetilde{cp}}$ increases to close to that of baseline L_{ce} (13.5 vs. 15.2), as shown in Table 1. Figure 7 shows that the gaps between accuracy and confidence of $L_{\widetilde{cp}}$ are amplified for almost all bins compared to L_{cp}. A simple remedy for $L_{\widetilde{cp}}$ is to post-process the outputs with temperature scaling. We found that the ECE of $L_{\widetilde{cp}}$ decreases from 13.5 to 12.4 with a temperature τ of 1.2.

7 Conclusion

We revisited the pathological behaviors of neural models in which they tend to be overconfident on inputs that appear meaningless to humans. We first analyzed the commonly used fact verification benchmarks with input reduction (Feng et al., 2018) and found that we could only shorten particular types of claims into a few tokens without changing the model's predictions. We explored various entropy regularization methods and also proposed our extensions. We found that regularizing the objective function with the reduced examples improves interpretability but deteriorates calibration. Training neural models that use more meaningful features while being well-calibrated is an important direction for future work.

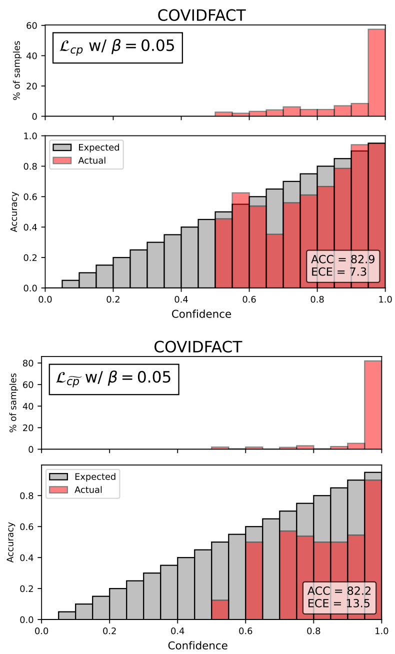

Figure 7: Confidence histograms and reliability diagrams for L_{cp} and $L_{\widetilde{cp}}$ on the COVIDFACT test set.

8 Limitations

Our work has several limitations. We focused on fact verification, which formulates the task sentence-pair (i.e., claim-evidence) classification. Our findings may hold for certain domains where the task format is similar (e.g., natural language inference or textual entailment recognition). We did not apply beam search on input reduction, which limits us from searching multiple versions of the reduced claims having the same length. We investigated three widely used regularization methods: temperature scaling, label smoothing, and the confidence penalty. However, other subsequent methods remain unexplored.

Acknowledgments

This work is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3), JST AIP challenge program, and MEXT KAKENHI Grants (21H04906), Japan.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Brandon Carter, Siddhartha Jain, Jonas W Mueller, and David Gifford. 2021. Overinterpretation reveals image classification model pathologies. In *Proceedings of Advances in Neural Information Processing Systems*, volume 34, pages 15395–15407.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330.

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.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Harold Jeffreys. 1946. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, volume 186, pages 453–461.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

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. *CoRR*, abs/1907.11692.

Clara Meister, Elizabeth Salesky, and Ryan Cotterell. 2020. Generalized entropy regularization or: There's nothing special about label smoothing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6870–6886.

Pakdaman Mahdi Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2901–2907.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Proceedings of Advances in Neural Information Processing Systems*, volume 32.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations*.

John Platt. 1999. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. 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 2116–2129.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267.

A Relationship between Jeffreys (J) divergence and Jensen–Shannon (JS) divergence

We can express the JS divergence between u and p as:

$$\mathrm{JS}(u \parallel p) = \frac{1}{2}\Big(\mathrm{KL}(u \parallel \frac{p+u}{2}) + \mathrm{KL}(p \parallel \frac{p+u}{2})\Big).$$

Both $\mathrm{JS}(u \parallel p)$ and $\mathrm{J}(u \parallel p)$ can be used as regularization functions. Following Lin (1991), the JS divergence is bounded by the J divergence:

$$\mathrm{JS}(u \parallel p) \leq \frac{1}{4}\mathrm{J}(u \parallel p).$$

Thus, the J divergence penalizes the loss more strongly than the JS divergence given the same β. We preliminarily examined the use of the JS divergence but found that it is not as effective as the J divergence in our task.

B Additional examples

Table 4 shows examples from the COVIDFACT test set where $L_{\widetilde{cp}}$ can successfully capture salient words.

Evidence:	IgG titers in SARS-CoV-infected healthcare workers remained at a significantly high level until 2015. All sera were tested for IgG antibodies with ELISA using whole virus and a recombinant nucleocapsid protein of SARS- CoV, as a diagnostic antigen. CONCLUSIONS IgG antibodies against SARS-CoV can persist for at least 12 years.
Label:	SUP
Claim:	**Long**-term persistence of igg antibodies in sars-cov infected healthcare workers
L_{cp}:	term persistence of igg antibodies in s - c infected healthcare workers
$L_{\widetilde{cp}}$:	**Long** term persistence igg antibodies in ars - ov infected
Evidence:	IgG titers in SARS-CoV-infected healthcare workers remained at a significantly high level until 2015. All sera were tested for IgG antibodies with ELISA using whole virus and a recombinant nucleocapsid protein of SARS- CoV, as a diagnostic antigen. CONCLUSIONS IgG antibodies against SARS-CoV can persist for at least 12 years.
Label:	REF
Claim:	**Pre**-term persistence of igg antibodies in sars-cov infected healthcare workers
L_{cp}:	term ars
$L_{\widetilde{cp}}$:	**Pre** - term persistence infected
Evidence:	Here, we utilize multiomics single-cell analysis to probe dynamic immune responses in patients with stable or progressive manifestations of COVID-19, and assess the effects of tocilizumab, an anti-IL-6 receptor monoclonal antibody.
Label:	SUP
Claim:	Single-**cell** omics reveals dyssynchrony of the innate and adaptive immune system in progressive covid-19
L_{cp}:	om ics reveals dy ss ynchron y of the innate and adaptive immune system in progressive cov id
$L_{\widetilde{cp}}$:	Single **cell** om ics reveals dy ss ynchron y of the innate adaptive immune progressive cov
Evidence:	Here, we utilize multiomics single-cell analysis to probe dynamic immune responses in patients with stable or progressive manifestations of COVID-19, and assess the effects of tocilizumab, an anti-IL-6 receptor monoclonal antibody.
Label:	REF
Claim:	Single-**brain** omics reveals dyssynchrony of the innate and adaptive immune system in progressive covid-19
L_{cp}:	ynchron immune
$L_{\widetilde{cp}}$:	Single **brain** om dy

Table 4: Examples of the original and reduced claims from the COVIDFACT test set where $L_{\widetilde{cp}}$ can retain the salient word, but L_{cp} fails. Both L_{cp} and $L_{\widetilde{cp}}$ correctly predict the label.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
9

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☑ A4. Have you used AI writing assistants when working on this paper?
Grammarly

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
6.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
6.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
6.3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
6.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.