Fine-Grained Visual Entailment Supplementary Material

Christopher Thomas*†, Yipeng Zhang*, and Shih-Fu Chang

Columbia University, New York, NY 10034, USA
{christopher.thomas, zhang.yipeng, sc250}@columbia.edu

1 AMR Annotation and Statistics

1.1 Annotation details

All our AMR strings used for annotation are produced by SPRING [1], which is a model that achieved recent SOTA performance on AMR semantic parsing, with a SMATCH score [3] of 83.

Our annotation interface is shown in Figure 1. The image, the hypothesis, the AMR graph, and the extracted KEs are shown to the annotators. The annotators are required to follow the definitions in PropBank [5] and the AMR 3.0 specifications [4] to annotate each KE (node or tuple), and provide the sample-level label. As mentioned in the main text, we provide the "opt-out" option for KEs. These opt-out KEs, in most cases, consist of nodes that require context (e.g., adjectives such as "`big`", numbers such as "`2`", time-indicators such as "`about-to`"). These KEs are not considered during evaluation.

*indicates equal contribution
†corresponding author

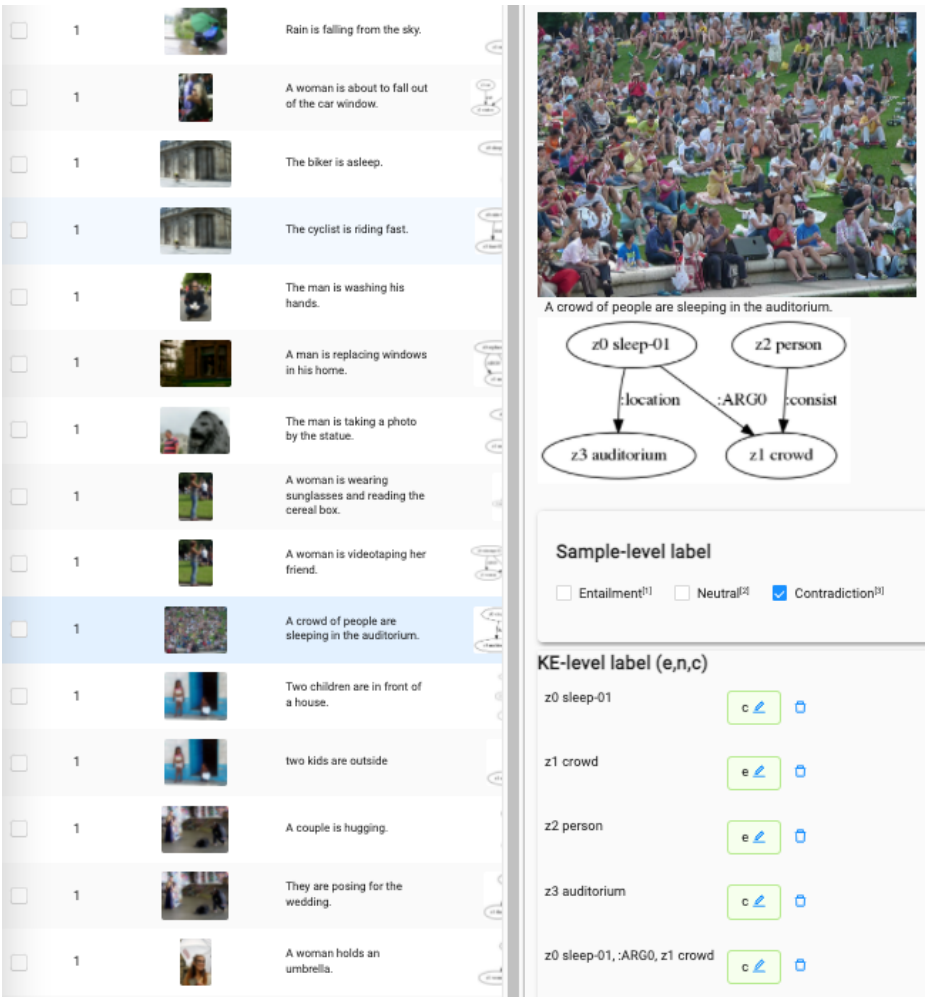


Fig. 1: Screenshot of our annotation interface.

1.2 Knowledge element distribution per category

In Figure 2, we show a breakdown of the distributions of both the knowledge element level annotations and the predictions by our method and the most competitive baseline on the KE-level. We show the normalized breakdown for each sample-level label (on the x-axis). For example, the three bars above the "contradiction" label on the x-axis indicate the distribution of KEs for those samples labeled contradiction. That is to say, of all the KEs in samples labeled contradiction, what percentage were labeled entailment (blue), neutral (purple), or contradiction (yellow).

In Figure 2a, we show the distribution of human annotated KEs (gold KEs) for each gold sample-level label. Note that the distribution for each sample-level label adds to one and that our gold labels exhibit no MIL violations (e.g., no neutral or contradiction KEs in samples labeled entailment). Significantly, we observe that entailed KEs make up a significant portion of both "neutral" and "contradiction" samples. For samples labeled neutral, there are more KEs labeled entailment than there are KEs labeled neutral (54% versus 46%). Similarly, for contradiction KEs, 58% of KEs are labeled contradiction and 41% are labeled entailed. If a method was perfectly accurate at predicting the sample-level label, one would get 54% of KEs wrong for neutral samples and 42% of KEs wrong for contradiction samples. This underscores the importance of our method of making fine-grained, KE-level predictions.

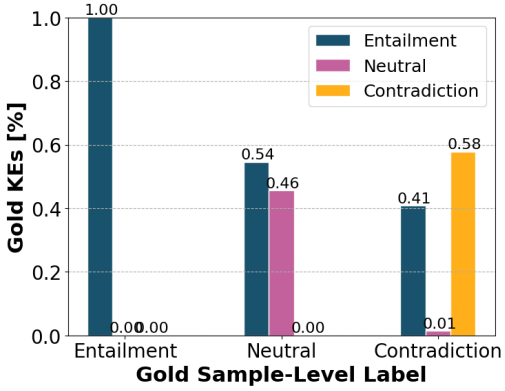
One surprising observation in Figure 2a is that there are very few "neutral" KEs in samples labeled contradiction. By definition, contradiction samples can contain entailed KEs (true claims about the image), neutral KEs (claims that could be true), and should contain at least one contradiction KE (a false claim). When Turkers were tasked with creating the SNLI dataset, they were prompted to "Write one alternate caption that is definitely a false description of the photo" for contradiction [2]. To do so, we observed that Turkers would often mention something specific truly in the the image (i.e., that was entailed), but then make an obviously false claim about it. Turkers usually didn't elaborate by adding additional "neutral" information into their hypotheses. However, because we neutral information *can* technically appear in a contradiction sample, we still allow our model to predict neutral KEs for contradiction samples. This lack of neutral KEs in contradiction samples is dataset specific and likely due to the way the dataset was constructed.

In Figure 2b, we show our method's KE-specific distributions across the three gold sample-level labels. We observe our distribution is close to the gold distribution (Figure 2a), with a few differences. For neutral, our method is overconfident about entailed KEs, predicting 68% of KEs as entailed, compared to the 54% which are actually entailed. In practice, our method confuses a number of truly neutral KEs as entailed. Note distinguishing some neutral KEs from entailed is often challenging because it requires one to distinguish whether a KE is true or merely could be true which is sometimes subjective. We observe that our method mistakenly predicts 9% of KEs in the neutral class as contradiction. For contradiction, we observe our method predicts 40% of KEs as entailed which

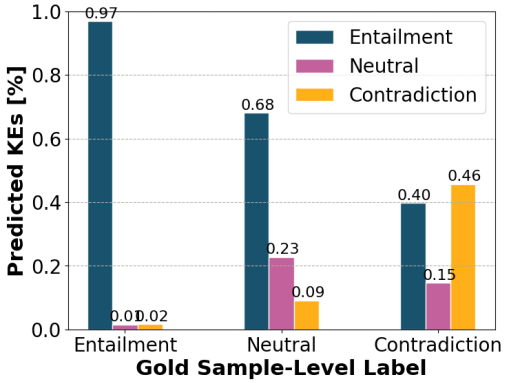
closely tracks the amount in the gold set (41%). We observe our model predicts 15% of KEs as neutral in contradiction. As stated above, we note that we could significantly improve our model's performance further on this dataset by enforcing a constraint that no neutral KEs were allowed in contradiction samples, but this would be a dataset-specific constraint. We instead chose to impose constraints consistent with the logical definitions that define the fine-grained visual entailment problem rather than tailor them to a particular dataset. Had these incorrectly predicted neutral KEs instead been predicted contradiction, our distribution for contradiction would be 61%, close to the 58% in the ground truth distribution.

In Figure 2c, we show the KE distribution of our most competitive baseline on the KE-level (VE+AMR→KE). We observe a significant degradation in performance compared to the gold distribution. For the entailment category, 10% of KEs are mistakenly predicted neutral. For the neutral category, the model predicts only 29% of KEs as entailed (54% in gold), while predicting 61% as neutral (46% in gold). This problem is most acute for the contradiction category where the model predicts 79% of KEs as contradiction (58% in gold) and only 7% as entailed (41% in gold). Without our KE-level constraints, the model's KE-level predictions are much more frequently the same as the the sample-level label than compared to our method or the gold labels, suggesting the predictions are not as semantically meaningful.

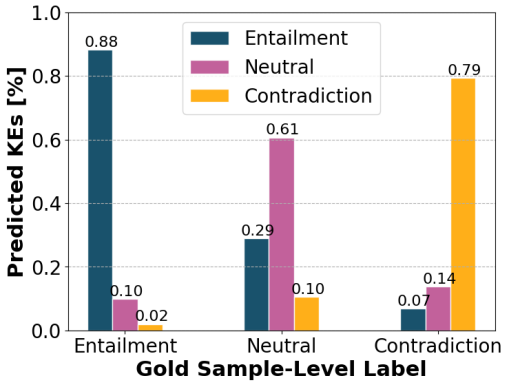
These distributions also explain why the baselines *appear* misleadingly strong at accurately predicting neutral and contradiction in Table 1 in the main text (left group of results), but then perform much worse overall. Because contradiction KEs only appear in contradiction samples and the model predicts the vast majority of KEs in contradiction samples as contradiction, the accuracy of the model on the set of contradiction KEs is very high (i.e., due to a high recall). However, the model is actually performing very poorly overall on contradiction samples (shown by the distribution), infrequently predicting any entailed KEs, when in actuality 41% of KEs are entailed. The same reasoning equally applies to the neutral case, where the baseline only predicts 29% of KEs as entailed in neutral samples, when in actuality 54% should be entailed.



(a) KE labels.



(b) KE predictions by our model.



(c) KE predictions by VE+AMR→KE.

Fig. 2: Distributions of (a) ground truth KE labels, (b) predicted KEs by our model within each sample category, and (c) predicted KEs by the VE+AMR→KE model within each sample category.

Table 1: Ablation of different loss weights used for training our model (Ours). For the results in this table, only the KE classifier f_{KE} is used. The best result per column is shown in bold and second best is underlined.

β_{CLS}	β_{KE}	β_{STRUC}	Acc_{ent}	Acc_{neu}	Acc_{con}	Acc_{node}	Acc_{tup}	$\text{Acc}^{\text{Relab.}}_{\text{KE}\rightarrow\text{CLS}}$	Acc_{KE}	$\text{Acc}_{\text{STRUC}}$
1	1	1	74.61	27.09	30.85	62.45	54.35	67.77	58.75	94.80
0.1	1	1	<u>84.75</u>	<u>30.64</u>	<u>44.68</u>	**71.84**	<u>64.10</u>	75.41	**68.30**	<u>96.46</u>
0.5	1	1	79.64	**31.29**	**62.76**	<u>69.79</u>	**66.02**	**80.73**	<u>68.07</u>	96.36
0.5	0.5	1	76.50	<u>30.64</u>	37.23	64.29	58.58	61.79	61.68	**98.40**
0.5	1	0	**88.87**	15.48	9.57	66.88	57.17	<u>79.73</u>	62.44	70.26

2 Ablation Study

We show ablation of different loss weights for our three losses in Table 1. First, competition exists between $\mathcal{L}_{\text{CLS}}$ and the other two losses, so we have to weigh $\mathcal{L}_{\text{CLS}}$ down for the model to achieve high KE performance. In other words, our model must trade off between focusing on the sample-level task and the KE-level task. This, however, does not undermine our performance on the sample level – setting β_{CLS} to 0.5 gives us a sample-level performance of 80.73% using only the KE predictions, best among all the methods studied in this paper. We also find that setting β_{CLS} too low (0.1) hurts the performance on the neutral and contradiction samples, as well as on the sample level (compared to $\beta_{\text{CLS}} = 0.5$). In general, these two models show comparable performance.

Structural constraints are essential for our model. If we remove the structural constraint, the model shows very low performance on both the neutral and contradiction samples; its structural accuracy drops to 70.26%, which means that the predictions might not be meaningful. On the other hand, putting less weight on $\mathcal{L}_{\text{CLS}}$ (0.5) and $\mathcal{L}_{\text{KE}}$ (0.5) and keeping $\beta_{\text{STRUC}} = 1$ yields lower performance on the KEs (61.68%), albeit reaching very high structural accuracy (98.4%). In summary, we find it beneficial to keep an equal weight on $\mathcal{L}_{\text{KE}}$ and $\mathcal{L}_{\text{STRUC}}$ and weighing $\mathcal{L}_{\text{CLS}}$ slightly lower.

3 Additional Implementation Details

3.1 AMR tokenization

The AMR string produced by SPRING [1] is separated by the new line character, ordered according to depth-first search (DFS). We use the SPRING model pretrained on AMR 3.0 and the `amrlib` library[1] for all AMR extractions.

We then remove the newline characters and the redundant token "/". Next, we merge the roles "`:opx`" into "`:op`" (`x` is an index) and roles "`:sntx`" into

[1] https://github.com/bjascob/amrlib

"`:snt`" because these indices do not affect the meaning conveyed. For example, "`:op1`" and "`:op2`" can be used interchangeably; on the other hand, "`:ARG0`" usually means the subject of a verb while "`:ARG1`" usually means the object, so the indices on "`:ARGx`" are essential for semantic representation. Note that the predicate labels (e.g., "`z0`") are also important because they provide information of cross-referencing.

Finally, We remove the predicate (node) indices (e.g., "`-01`" in "`walk-01`") used for disambiguating different meanings for predicates. This is because we rely on the transformer for disambiguation, given their capability of disambiguating different meanings for words in the English corpus. We also find such approach work better empirically.

These changes not only shorten the AMR sequences, but also create more instances of "`:op`", "`:snt`", and the predicate tokens, which helps learning. We add all the edge tokens, the "`amr-unknown`" predicate, as well as the role-inversion indicator "`-of`" into the vocabulary and initialize each newly added token's embedding as the average embedding of their word pieces (pretrained by Oscar+). We do not add new tokens for predicates other than "`amr-unknown`".

3.2 Other implementation details

Oscar+ is pretrained on three sequences (object features, tags, text), with the tag sequence and text sequence using distinct token type embeddings. When we add another sequence, the AMR sequence, we create a new token type embedding and initialize it with the pretrained token type embedding of the text. All the parameters in the model are finetuned on our task.

We follow Oscar+ [6] and uses object features of dimension 2054. Among it, 2048 are the extracted region features; four dimensions are the bounding box coordinates (left, top, right, bottom) normalized by image size; the rest two dimensions are the normalized width and height of the object. The object detector's confidence threshold is set to 0.2, as in Oscar+ [6].

Our losses are computed on KEs. However, in rare cases when the sequence is too long, some ending KE tokens could get truncated. If a sample with neutral or contradiction label has such tokens, we do not enforce KE or structural losses on them so as not to introduce false signals.

Before summing up the losses, we take the average of $\mathcal{L}_{\text{CLS}}$ and $\mathcal{L}_{\text{KE}}$ across each sample and $\mathcal{L}_{\text{STRUC}}$ across each relation. Since we do not have fine-grained KE labels for the training set, the checkpoint that performs the best on the sample level using f_{CLS} (on the validation set) is saved for evaluation.

4 Additional Qualitative Results

We show additional qualitative results in Figure 3. As in our main text, the color of the edges (representing the tuple) and nodes indicate the predicted label. A dotted line indicates an incorrectly predicted KE. We observe that our model is able to accurately identify the source of inconsistencies and logically reason about the relationship of the image to each hypothesis.

In Figure 3 (a), we observe the model identifies that there is no dog and that no dog is skiing. In (b), the model identifies that no one is walking, is unsure about the location, but does detect a crowd of people. In (c), the model identifies no one is sleeping, but correctly determines there are two police officers and there is a street. In (d), several tuples and the have-rel-role node can't be predicted without additional context, but the model accurately determines that the wife relationship is unknown. In (e), our model correctly concludes the action (work) is entailed, that there are two women, and they are working in a factory. Note that the model is neutral as to whether the factory is big, as is the human annotator. In (f), we see our model mistakenly concludes that the people are not "exiting." We see that some individuals are standing or staring, but whether they are actually exiting is unclear. The model's understanding of exiting may be that exiting only occurs when someone is walking out a door, for example, and thus the model predicted contradiction. In (g), we see a another mistake of our method. We observe that our model believes someone is "playing", but is unclear if it is the man who is playing. The model is unclear as to whether music is present or if it is being played. However, the model correctly identifies that the man is elderly and can't determine if the music is original. The model may have become confused in this example by the unusual instrument and the position of objects. Lastly, in (h) we observe one of the rare structural violations from our model. The model is unclear as to whether "buying" is occurring, but predicts "person buying" and "waiting to buy" as entailed, resulting in a bottom-up structural violation.

Fig. 3: Additional qualitative results showing our KE-level predictions on AMR graphs. Nodes and edges (representing tuples) are colored based on their predicted label (ent, neu, con, opt-out). Wrong predictions are denoted by dashed lines. We drop predicate labels (e.g., `z0`) for brevity.

References

1. Bevilacqua, M., Blloshmi, R., Navigli, R.: One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In: Proceedings of AAAI (2021)
2. Bowman, S., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 632–642 (2015)
3. Cai, S., Knight, K.: Smatch: an evaluation metric for semantic feature structures. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 748–752 (2013)
4. Knight, K., Badarau, B., Baranescu, L., Bonial, C., Bardocz, M., Griffitt, K., Hermjakob, U., Marcu, D., Palmer, M., O'Gorman, T., Schneider, N.: Abstract meaning representation (AMR) annotation release 3.0. `https://catalog.ldc.upenn.edu/LDC2020T02` (2020)
5. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Computational linguistics **31**(1), 71–106 (2005)
6. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., Gao, J.: Vinvl: Revisiting visual representations in vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5579–5588 (2021)