GSRFormer: Grounded Situation Recognition Transformer with Alternate Semantic Attention Refinement

Zhi-Qi Cheng  Carnegie Mellon University  zhiqic@cs.cmu.edu
Qi Dai  Microsoft Research  qid@microsoft.com
Siyao Li  Carnegie Mellon University  siyaol@andrew.cmu.edu
Teruko Mitamura  Carnegie Mellon University  teruko@cs.cmu.edu
Alexander G. Hauptmann  Carnegie Mellon University  Alex@cs.cmu.edu

ABSTRACT
Grounded Situation Recognition (GSR) aims to generate structured semantic summaries of images for "human-like" event understanding. Specifically, GSR task not only detects the salient activity verb (e.g., buying), but also predicts all corresponding semantic roles (e.g., agent and goods). Inspired by object detection and image captioning tasks, existing methods typically employ a two-stage framework: 1) detect the activity verb, and then 2) predict semantic roles based on the detected verb. Obviously, this illogical framework constitutes a huge obstacle to semantic understanding. First, pre-detecting verbs solely without semantic roles inevitably fail to distinguish many similar daily activities (e.g., offering and giving, buying and selling). Second, predicting semantic roles in a closed auto-regressive manner can hardly exploit the semantic relations among the verb and roles. To this end, in this paper we propose a novel two-stage framework that focuses on utilizing such bidirectional relations within verbs and roles. In the first stage, instead of pre-detecting the verb, we postpone the detection step and assume a pseudo label, where an intermediate representation for each corresponding semantic role is learned from images. In the second stage, we exploit transformer layers to unearth the potential semantic relations within both verbs and semantic roles. With the help of a set of support images, an alternate learning scheme is designed to simultaneously optimize the results: update the verb using nouns corresponding to the image, and update nouns using verbs from support images. Extensive experimental results on challenging SWiG benchmarks show that our renovated framework outperforms other state-of-the-art methods under various metrics\(^1\).

CCS CONCEPTS
• Computing methodologies → Vision and Language.

KEYWORDS
Grounded Situation Recognition; Transformer Framework

\(^1\)Code is available at https://github.com/zhiqic/GSRFormer

ACM Reference Format:
Zhi-Qi Cheng, Qi Dai, Siyao Li, Teruko Mitamura, and Alexander G. Hauptmann. 2022. GSRFormer: Grounded Situation Recognition Transformer with Alternate Semantic Attention Refinement. In Proceedings of the 30th ACM International Conference on Multimedia (MM ’22), Oct. 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3503161.3547943

1 INTRODUCTION
Understanding complex events in a way that obeys human cognitive habits is one of the core tasks of computer vision and multimedia. As shown in Figure 1, “human-like” event understanding goes beyond traditional object- and action-centric detection and recognition tasks [11, 13, 24, 30, 31, 39]. Different from image captioning [9, 32, 43, 78] and scene graph generation [20, 67, 68, 71], which use natural language or object-relational graphs to describe scenes, human-friendly event understanding must be event-centric, that is, not only to identify what activities happened, but also to recognize how objects participate in activities, i.e., answer questions like “who is doing what with some tools at someplace.”

To meet the demands of “human-like” event understanding, inspired by previous research on semantic role labeling [21, 45, 53, 59] in text, Grounded Situation Recognition (GSR) [47, 73] is proposed for event understanding in the multimedia domain. As illustrated in Figure 1, GSR not only detects the salient activity (verb) in the image (e.g., sprinkling), but also recognizes all semantic roles (e.g., agent is man, source is cup). To further determine semantic roles in images, GSR also provides visually grounded information (i.e., bounding boxes) for noun entities. From the perspective of the theory of frame semantics [26], GSR task can be considered as a multimedia extension to earlier lexical databases such as FrameNet [3] and PropBank [33]. By describing activities with verbs and grounded semantic roles, GSR can provide a visually grounded

Figure 1: Given an input image, Grounded Situation Recognition (GSR) not only detects the salient verb category (e.g., sprinkling), but also predicts all corresponding semantic roles for sprinkling, such as agent: man, item: spice, and source: cup, etc.
verb-frame, which benefits many downstream scene understanding tasks, such as information retrieval [15, 16, 44, 56], question answering [1, 7, 35, 67], recommended system [10, 12, 14, 55], and multimedia understanding [29, 51, 77, 78].

To sum up, the essence of GSR task is using semantic relations to generate verb-frames for event understanding. However, inspired by object detection and image captioning, almost all current GSR methods [19, 36, 42, 47, 54] adopt a two-stage framework. As shown in Figure 3 (a-b), two-stage framework 1) first blindly pre-detects verbs to reduce the search space, and then 2) predicts the semantic roles in an auto-regressive (RNN) or parallel (Transformer) manner depending on the detected verbs. Such two-stage frameworks obviously neglect the semantic relations among verbs and semantic roles. On the one hand, pre-detecting the verb without noun entities inevitably fails to distinguish some similar daily activities. For example, as shown in Figure 2 (left), it is hard to distinguish similar verbs (e.g., buying and giving) without the help of any noun entities. On the other hand, based on pre-identified verbs, applying auto-regression in a closed space would accumulate errors and thus miss the semantic relationships. As shown in Figure 2, once verb buying is wrongly predicted as giving in the first stage, the semantic roles payment: credit and place: shop could be neglected in the second stage.

Similar to our starting point, previous work CoFormer [18] and SituFormer [65] also argue that the existing two-stage framework is problematic. As shown in Figure 3 (c), a three-stage framework is thus proposed to further optimize verbs with predicted nouns. Following traditional two-stage works, the first two stages perform the verb and noun entities detection, respectively. Then, in the third stage, the predicted noun entities are used to refine the verb. However, such disentangled framework still has the following flaws. First, the bidirectional semantic relations between verbs and noun entities cannot be fully exploited. They either use the noun roles to refine the verbs (SituFormer), or only use verbs to refine the noun roles (CoFormer) while ignoring each other. Second, the framework is redundant. It has two parallel transformer verb and noun detectors, but does not learn semantic relations between them during the encoding phase. Third, the refinement process is not scalable. It can be treated as a one-time noun-to-verb optimization, which apparently cannot be expanded.

To address these issues, we focus on how to exploit such bidirectional semantic relations within verbs and noun entities, which can constrain each other. Rather than explicitly pre-detecting the verbs at the first stage, we postpone decision verbs, thus simply assuming a pseudo-category and learning an intermediate representation for each noun entity. We then devise an iterative framework to capture the semantic relations among verbs and nouns and alternately learn their features. As such, we streamline the redundant structures and make them flexibly handle semantic roles in parallel.

Technically, our proposed method, called GSRFormer, is built based on the transformer structure [62] due to its parallel processing capability. As shown in Figure 4, GSRFormer adopts a two-stage architecture that consists of an encoder and a decoder. In the encoder part, we first utilize stacked Multi-Head Attention (MHA) layers to learn the feature of the verb. By assuming a pseudo category of the verb, we further learn the intermediate representations of the corresponding noun entities from the image. In the decoder part, MHA layers are employed to mine the implicit relations among both verbs and nouns. By leveraging a set of support images, the model alternately optimizes the verbs and nouns in a loop: update the features of verbs using nouns, and vice versa. Our framework successfully learns the semantically rich representations for both verbs and nouns and thus performs accurate recognition. To conclude, our contributions are mainly three folds:

- We reveal the problems of existing frameworks and point out that learning the bidirectional semantic relations is the core for accurate role recognition.
- We propose a two-stage framework with transformer structures to iteratively refine activity verbs and noun entities. It flexibly mines the potentially open semantic relations within verbs and nouns and alternately updates their features.
- Extensive experiments on challenging SWiG benchmarks fully demonstrate that our proposed framework outperforms other state-of-the-art methods under various metrics.

2 RELATED WORK

History of SR to GSR tasks. Although deep learning achieves satisfactory performance in basic vision tasks such as action recognition [6, 30, 39, 61, 64, 70, 79] and object detection [5, 11, 24, 40, 48, 58], they still cannot fully understand events in natural scenes. To address this problem, image captioning [1, 23, 49, 52, 69, 75] and scene graph generation [8, 20, 37, 66, 68, 71, 76] attempt to reason and describe scene content through natural language or relational graphs of objects. However, these efforts have still failed to understand events consistent with human cognition, i.e., identifying what happened and who was involved in what roles.

In this context, Yatskar et al. [73] first proposed Situation Recognition (SR) task and annotated imSitu dataset as the benchmark. However, the original SR task cannot point to where the involved noun entities are located in the image. To further address the visual grounding of the entities, Pratt et al. [47] redefines Grounded Situation Recognition (GSR) task and proposes SWiG dataset by providing bounding box annotations on imSitu dataset. GSR task can be seen as a further extension of SR task.
Challenges of GSR tasks. The challenges of GSR task are to handle the semantic relations in the scene. Yatskar et al. [73] proposed a conditional random field (CRF) model in the initial phase. However, follow-up work [72] pointed out that CRF method cannot effectively utilize semantic relations. Since then, a lot of works have started to investigate how to model the relations among semantic roles. The previous technical routes mainly include Recurrent Neural Network (RNN) [42, 47, 63, 63], Graph Neural Network (GNN) [36, 54] and Relational Reasoning [4, 22], etc.

Transformers in GSR task. After great success in NLP tasks, transformer structure [62] was introduced to solve various computer vision problems, including image generation [17, 46], image recognition [25, 41, 60], object detection [5, 80], object segmentation [74], image captioning [23] and scene graph generation [20], etc. To further exploit the strength of the transformer in GSR task, Cho et al. [19] proposed the first transformer framework (GSRTR) by replacing the object queries in transformer object detector (DETR [80]) with semantic role queries. Wei et al. [65] recently proposed SituFormer, which uses two transformer-based verb and noun detectors to improve the performance. In addition, Cho et al. [18] proposed CoFormer that tends to leverage the semantic relations between the verb and noun roles to refine the prediction.

Problem of framework. To elaborate, as shown in Figure 3, almost all existing GSR methods [4, 19, 22, 47, 47, 54, 63] adopt a two-stage framework based on RNN or transformer. In RNN structure, Pratt et al. [47] proposes a Joint Situation Localizer (JSL) model, which consists of a verb classifier in the first stage and an RNN-based object detector in the second stage. In transformer structure, GSRTR [19] uses a transformer encoder to detect verbs in the first stage and a transformer decoder to predict semantic roles in the second stage. These two-stage frameworks evidently cannot exploit the semantic relations. First, recklessly detecting verbs in the first stage will inevitably misidentify similar activities. Second, the closed auto-regressive strategy of the second stage not only fails to correct misrecognized verbs, but leads to more mispredicted semantic roles. Although recent works (SituFormer [65] and CoFormer [18]) are also aware of framework issues and adopt a three-stage framework to refine prediction results from coarse to fine, they are still unable to optimize the results with bidirectional semantic relationships (i.e., from both verbs and nouns).

3 PROPOSED METHOD

3.1 Problem Formulation

Definition of GSR task. Given an image I, GSR aims to generate a structured verb frame \(\mathcal{F}_o = \{v, \mathcal{R}_o\}\). As shown in Figure 1, GSR not only recognizes the salient verb \(v \in \mathcal{V}\), but also detects all corresponding semantic roles \(\mathcal{R}_o = \{(r, n_r, b_r)\} \) for \(r \in \mathcal{R}_o\), where \(\mathcal{R}_o = \{r_1, ..., r_m\}\) is the set of role types for verb \(v\). For instance, the verb-frame in Figure 1 can be instantiated as \(\mathcal{F}_o = \{\text{Sprinkling}, \{(\text{Agent, Man}, \square), (\text{Item, Spice, \square}), (\text{Source, Cup, \square}), (\text{Destination, Pan, \square}), (\text{Place, Kitchen, \theta_p})\}\}\). Semantic roles \(\mathcal{R}_o\) is a collection of triples, where each role contains the role type \(r\), the noun entity \(n_r \in \mathcal{N}\) and the corresponding bounding box \(b_r \in \mathbb{R}^4\). Note that not all semantic roles have corresponding nouns and bounding boxes, i.e., \(n_r\) or \(b_r\) can be equal to \(\emptyset\).

Problems of existing frameworks. Inspired by object detection and image captioning, as shown in Figure 3, the existing GSR methods [19, 36, 42, 47, 54] widely adopt a two-stage framework, i.e., 1) identifying salient verbs \(v\) in the first stage, and 2) detecting the corresponding semantic roles \(\mathcal{R}_o\) in the second stage:

\[
P(\mathcal{F}_o | I) = P(v | I) P(\mathcal{R}_o | v, I).
\]

There are multiple highly similar activities in GSR tasks, such as buying and giving, as shown in Figure 2. This two-stage framework is inherently unable to utilize the noun entities in the first stage to distinguish similar verbs, let alone misidentify semantic roles due to the verb prediction errors accumulated in the second stage.

To address these issues, Wei et al. [65] recently proposed a three-stage framework (SituFormer) as:

\[
P(\mathcal{F}_o | I) = P(\mathcal{V}_c | I) P(\mathcal{R}_o | \mathcal{V}_c, I) P(v | \mathcal{R}_o) P(\mathcal{V}_c | \mathcal{R}_o, \mathcal{V}_c, I).
\]

where the main idea is to refine verb predictions in a coarse-to-fine manner. As shown in Figure 3 (c), it uses the first two stages to identify a set of candidate verbs \(\mathcal{V}_c\) and corresponding noun entities \(\mathcal{R}_o\). Then the third stage uses a ranking loss to refine the verb \(v\) with a set of support images through similarity retrieval.

These predefined verb frames are filtered from PropBank [33] or FrameNet [3, 27].
As shown in Figure 4, GSRFormer consists of a transformer-based multi-head attention modules to learn the intermediate features for verbs and semantic roles. Transformer decoder consists of four parts. 1) A set of similar activities (verbs) are retrieved using features from the encoder. 2) A multi-head attention layer is used to compute the messages $p$, thus capturing semantic relations among verbs and nouns. 3-4) The features of nouns and verbs are updated alternately with the computed messages. Note that we stack multiple decoder layers for iterative optimization (refinement).

Similarly, Cho et al. [18] also proposed a three-stage framework (CoFormer). From the framework perspective, CoFormer is similar to SituFormer. 1) The first stage coarsely predicts noun roles (Glance). 2) The second stage uses the predicted noun roles to help predict the verb (Gaze-Step1). 3) The third stage refines the candidate noun roles in Gaze-Step1 using the predicted verb (Gaze-Step2).

Obviously, these revised frameworks are still unreasonable due to the following issues. 1) It uses only noun entities to refine the verb, but totally ignores semantic constraints in the opposite direction (i.e., from verbs to noun entities). 2) Although it has two redundant transformer-based noun and verb detectors, it does not learn the semantic relations between them at the encoding stage. 3) The coarse-to-fine refinement process can only be done once, and it is impossible to iteratively optimize both verbs and nouns.

3.2 Overview of GSRFormer

To solve these problems, we reconstruct a framework (GSRFormer) with transformers to fully exploit the semantic relations of GSR tasks. We reformulate GSR task as,

$$
P(F_v|I) = P(\hat{v}, H_v|I) \cdot P(\hat{v}, R_v|H_v, I),$$

where $\hat{v}$ is the assumed pseudo verb category, and $H_v$ is the intermediate feature for the verb and associated roles. The goodness of $H_v$ is that it yields the essence of the GSR task, which is to extract comprehensive semantic relations and then iteratively refine the results. As shown in Figure 4, GSRFormer consists of a transformer-based encoder and decoder, respectively. In the first step (Sec. 3.3), instead of pre-detecting verbs, the transformer encoder learns intermediate representations $H_v$ of verbs and corresponding semantic roles from images, respectively. In the second step (Sec. 3.4), we first use the features obtained in the encoder to retrieve the representations of Top-K similar verbs as the support verb set $\{H_v\}_s$. To mine the various semantic relations, the neural message passing mechanism [28] is then used to flexibly associate relevant relations to each verb or role, thus effectively updating their representations. Finally, we take full advantage of the transformer structure to perform multiple iterations to refine verbs $v$ and semantic roles $R_v$.

3.3 Transformer-based Encoder

The transformer encoder is devised to learn intermediate representations of verbs and semantic roles from images using two multi-head attention modules, respectively.

**Representation of verbs.** As shown in Figure 4, given an image $I$, CNN backbone first extracts the feature map $X_{img} \in \mathbb{R}^{d \times h \times w}$. Since the input to the transformer encoder is a sequence of tokens, a $1 \times 1$ convolutional layer and a flatten operator are used to convert the $X_{img}$ into a sequence of “visual” tokens $[f_1, ..., f_n]$, where each token $f_i \in \mathbb{R}^d$ is compressed as a $d$-dim visual feature. Inspired by the classification token used in ViT [25], we initialize a learnable verb token $f_0 \in \mathbb{R}^d$ to stand for verb $v$. Then all visual and verb token sequences are fed into the first encoding module as,

$$e_{v_0}, e_{v_1}, ..., e_{v_{n_v}} = \text{MHA}_v^\text{img}(f_0, f_1, ..., f_n, \oplus E_{pos}),$$

where $\oplus$ is element-wise addition and $E_{pos}$ is positional embedding used to distinguish relative positions in the sequence. $\text{MHA}_v^\text{img}$ is a set of stacked multi-head self-attention blocks. As shown in Figure 4, each block consists of a multi-head self-attention layer and a feed-forward network, and the layer normalization [2] (Add & Norm) is used before both of them. The role of $\text{MHA}_v^\text{img}$ module is to use multi-head self-attention to learn the intermediate representation of the verb from image features. The output can be divided into optimized 1) image features $E_{img} = \{e_{v_1}, ..., e_{v_{nv}}\} \in \mathbb{R}^{d \times h \times w}$ and 2) verb feature $e_v \in \mathbb{R}^d$. Here verb feature $e_v$ is fed to a classifier to determine a pseudo verb category $\hat{v}$. After obtaining $\hat{v}$, we can fetch the corresponding semantic roles $R_v$ and initialize them to a sequence $[r_1, ..., r_n]$, where each initialized role $r_i \in \mathbb{R}^d$ is a $d$-dim visual feature.

$E_{pos}$ has the same dimension as sequences of visual and verb tokens. For more implementation details, see original papers [19, 62].
Representation of semantic roles. We further learn intermediate representations of corresponding semantic roles plus verb $\mathcal{H}_0 = \{h^0_v, h^0_{v-1}, \ldots, h^0_{v-n_m}\}$ from images. Specifically, $E_{img}$ and $e_v$ are used as the input to the second encoding module as,

$$h_v^{(0)}, h_{v-1}^{(0)}, \ldots, h_{v-n_m}^{(0)} = \text{MHA}_{\text{img}}^{\text{roles}}(E_{img} \oplus E_{img} \oplus e_v, r_1, \ldots, r_m) \tag{5}$$

where we concatenate verb feature $e_v$ and corresponding semantic role embeddings $\{r_1, \ldots, r_m\}$ as query. As shown in Figure 4, similar to the previous encoding module, MHA$_{\text{img}}^{\text{roles}}$ is also a set of stacked multi-head attention blocks. The output representation $\mathcal{H}_0$ is then utilized in the decoder part. After GSRFormer encoder, we assume that the feature $\mathcal{H}_0$ has captured the semantics from the image.

3.4 Transformer-based Decoder

The transformer decoder aims to utilize the semantic relations among a set of similar verbs and corresponding semantic roles to simultaneously refine their features. While transformer itself is a powerful model that can mine the implicit relations of all pairwise interactions, it still struggles to leverage additional domain-specific knowledge. In our framework, we expect to exploit a set of support verbs (excluding their corresponding nouns) to profit the noun refinement, while only using the nouns in one single image to refine its verb. This a priori knowledge can hardly be included when directly applying the transformer since it considers ALL pairwise relations. To this end, we borrow ideas from the neural refinement, while only using the nouns in one single image to leverage additional domain-specific knowledge. In our framework, we expect to exploit a set of support verbs set $\mathcal{V}_s$ and corresponding semantic roles plus verb $\mathcal{H}_0 = \{h^0_v, h^0_{v-1}, \ldots, h^0_{v-n_m}\}$ learned from the encoder to retrieve the features of top-K similar verbs $\{\mathcal{H}_0\}_s = \{\mathcal{H}_{t_0}, \ldots, \mathcal{H}_{t_K}\}$ as support verbs set,

$$\{\mathcal{H}_0\}_s = \arg \max_{\mathcal{H}_{t_j}, v_j \in \mathcal{D}_F} S(\mathcal{H}_0, \mathcal{H}_{t_j}) \tag{6}$$

where $\mathcal{D}_F$ is the set of all training images. The similarity score $S(\cdot, \cdot)$ is the average cosine similarity of semantic roles as,

$$S(\mathcal{H}_0, \mathcal{H}_{t_j}) = \frac{1}{m} \sum_{i=1}^{m} \text{sim}(h_{v-n_i}^{(0)}, h_{v-j-n_i}^{(0)}), \tag{7}$$

where $\text{sim}(\cdot)$ is cosine similarity. $h_{v-n_i}$ and $h_{v-j-n_i}$ are the noun entity features for the verbs $v$ and $v_j$, respectively.

Semantic relation message computation. Given retrieved support set $\{\mathcal{H}_0\}_s$, we aim to update the feature representation of each element in $\mathcal{H}_0$ by leveraging the relations between or within $\mathcal{H}_0$ and $\{\mathcal{H}_0\}_s$. Following the standard message passing paradigm, we compute one message $\psi$ for each involved verb and noun in one image by a multi-head attention (MHA) layer as,

$$\mathcal{H}_{\text{values}}^{\text{roles}}(h_v^{(t)}, r_{v-n_1}, \ldots, r_{v-n_m}) = \text{Agg}(\mathcal{H}_{\text{values}} \oplus \mathcal{H}_{\text{values}} \oplus \mathcal{H}_{\text{values}} \oplus \mathcal{H}_{\text{values}}) \tag{8}$$

where $\oplus$ is element-wise addition, and $t$ implies the $t$-th iteration. $h_v^{(t)}$ and $h_{v-n_i}^{(t)}$ are the learned verb and noun entity features, respectively. Here we replace the positional encoding in the original MHA layer with the semantic role embedding $r_{v-n_i}$. Thus MHA$_{\text{values}}^{\text{roles}}$ is actually a multi-head self-attention model. We perform the relation message computation for the entire support set. The obtained verb message $\psi_v^{(t)}$ and the noun message $\psi_{v-n_i}^{(t)}$ contain all the semantic information within a single image. Below we will consider semantic relations in multiple verbs (i.e., support verbs set).

Refine noun entity with verbs. We utilize the semantic relations (messages) from the verbs of support set to refine the noun entities.

To update the representation of a noun entity $h_{v-n_i}^{(t)}$, a single update message $\psi_{v-n_i}^{(t)}$ is computed by aggregating the messages of support verbs $\{\psi_{v-n_i}^{(t)}\}$ as Eq. 9. The aggregation function $\text{Agg}(\cdot)$ can be any permutation-invariant function (e.g., element-wise sum and max), and here we employ a gated update function [38]. After the messages are fused, a transformer sublayer (FFN and LN) is used to updated representation with residual connection:

$$\psi_{v-n_i}^{(t)} = \text{LN}(\mathcal{H}_{v-n_i}^{(t)}) = \text{Agg}(\psi_{v-n_i}^{(t)} | 1 \leq k \leq K), \tag{9}$$

$$\psi_{v-n_i}^{(t)} = \text{LN}(\mathcal{H}_{v-n_i}^{(t)} + \psi_{v-n_i}^{(t)}), \tag{10}$$

$$\mathcal{H}_{v-n_i}^{(t+1)} = \text{LN}(\mathcal{H}_{v-n_i}^{(t)} + \text{FFN}(\psi_{v-n_i}^{(t)})). \tag{11}$$

where LN$(\cdot)$ is layer normalization [2] and FFN$(\cdot)$ is a feed-forward neural network (commonly with one large intermediate layer).

Refine verb with noun entities. Similarly, when updating the verb feature, we utilize the messages of nouns only from the single associated image, as shown in Eq. 12-14,

$$\psi_{v-n_i}^{(t)} = \text{Agg}(\psi_{v-n_i}^{(t)} | 1 \leq i \leq m), \tag{12}$$

$$\psi_{v-n_i}^{(t)} = \text{LN}(\psi_{v-n_i}^{(t)} + \psi_{v-n_i}^{(t)}), \tag{13}$$

$$\mathcal{H}_v^{(t+1)} = \text{LN}(\psi_v^{(t)} + \text{FFN}(\psi_v^{(t)})). \tag{14}$$

Note that the above two refining processes can be accomplished alternatively. Unlike the previous work [65], which only optimized from rough noun entities to verbs, our framework takes full advantage of the flexibility of the transformer structure to efficiently perform multiple refinement iterations.

After completing the refinement of noun entities and verb features for $T$ iterations, we employ a lightweight MLP over the verb and noun entity features, respectively, to achieve the classification of verbs and the regression of noun entities with bounding boxes,

$$v = \text{MLP}(h_v^T), \tag{15}$$

$$\{n_i, b_i\} = \text{MLP}(h_v^T). \tag{16}$$

We discuss the training process in detail in the following section. In the ablation studies, we discuss the effect of the number of loops and refinement order on the results in detail.
Top-1-Verb | verb  | value  | val-all | grnd | grnd-all |
|--------|-------|--------|--------|------|---------|
| CRF [73] | 32.25 | 24.56  | 14.28  | –    | –       |
| CRF+DataAug [72] | 34.20 | 25.39  | 15.61  | –    | –       |
| VGG+RNN [42] | 36.11 | 27.74  | 16.60  | –    | –       |
| FC-Graph [36] | 36.93 | 27.52  | 19.15  | –    | –       |
| CAQ [22] | 37.96 | 30.15  | 18.58  | –    | –       |
| Kernel-Graph [54] | 43.21 | 35.18  | 19.46  | –    | –       |

Methods for Situation Recognition

| Ground-Truth-Verb | value | val-all | grnd | grnd-all |
|-------------------|-------|--------|------|---------|
| CRF+DataAug [72] | 58.64 | 42.68  | 22.75 | –       |
| VGG+RNN [42] | 62.21 | 46.72  | 25.66 | –       |
| FC-Graph [36] | 63.11 | 47.09  | 26.48 | –       |
| CAQ [22] | 64.99 | 45.23  | 29.98 | –       |
| Kernel-Graph [54] | 68.5  | 56.32  | 30.56 | –       |

Methods for Grounded Situation Recognition

| ISL [47] | 38.83 | 30.47  | 18.23 | 22.47 | 7.64 |
| JSL [47] | 39.60 | 31.18  | 18.85 | 25.03 | 10.16 |
| GSRTR [19] | 41.06 | 32.52  | 19.63 | 26.04 | 10.44 |
| SituFormer [65] | 44.32 | 35.35  | 22.10 | 29.17 | 15.33 |
| CoFormer [18] | 44.41 | 35.87  | 22.47 | 29.37 | 12.94 |
| GSRFormer (ours) | 46.64 | 37.69  | 23.58 | 31.61 | 14.42 |

| Methods for Grounded Situation Recognition | Top-1-verb | Top-5-verb |
|-------------------------------------------|------------|------------|
| ISL [47] | 65.74 | 50.29 |
| JSL [47] | 67.71 | 52.06 |
| GSRTR [19] | 69.46 | 53.69 |
| SituFormer [65] | 72.98 | 57.58 |
| CoFormer [18] | 73.43 | 58.75 |
| GSRFormer (ours) | 73.77 | 59.49 |

### 3.5 Training Objectives

We use the same data augmentation and batch training strategy as previous work [19]. The training details for the encoder and decoder are as follows.

**Training of encoder.** We use the cross-entropy loss function to train the encoder to obtain the pseudo-verb category as,

\[ L_{verb} = L_{CE}(\phi^d, \tilde{v}), \]

where the ground-truth verb category is denoted as \( \phi^d \) and the predicted pseudo-verb category is \( \tilde{v} \). Note that the first multi-head attention module of the encoder performs gradient back-propagation only when training the encoder, and does not participate in parameter updates when training the decoder.

**Training of decoder.** When training the decoder, we need to optimize the categories of verbs and nouns as well as the bounding boxes of nouns. Following previous work [19, 65], the losses of the decoder are calculated as,

\[ L_{verb} = L_{CE}(\phi^d, v), \]

where \( L_{verb} \) is the cross-entropy loss for noun classification. \( L_{box} \) consists of the generalize IOU loss [50] and the L1 regression loss.

### 4.1 Experimental Settings

#### Datasets

Our experiments are carried out on the challenging SWiG benchmark [47]. SWiG dataset builds on the original Situation Recognition (SR) imSitu dataset [73] by adding bounding box (bbox) annotations for all visible semantic roles (63.9% of roles have bbox annotations). Since each image in imSitu is annotated with three verb frames by three annotators, SWiG contains 126,102 images with 504 verbs and 190 semantic role types, and each verb is followed by 1 to 6 semantic roles (3.55 on average). We followed the official splits to construct the training/validation/testing set with sizes of 75K/25K/25K, respectively.

**Evaluation metrics.** We use the same five evaluation metrics as Pratt et al. [47], including 1) **verb**: the accuracy of verb prediction, 2) **value**: the accuracy of noun prediction for each semantic role, 3) **val-all**: the accuracy of noun prediction for the whole semantic role set, 4) **grnd**: the accuracy of noun prediction with correct grounding (bbox) for each semantic role, 5) **grnd-all**: the accuracy of noun prediction with grounding (bbox) for the whole semantic role set. Note that we consider a grounding is correct if the IoU between the predicted and ground-truth bbox is above 0.5. Meanwhile, we report the above metrics in three evaluation settings: 1) **Top-1-verb**, 2) **Top-5-verb** and 3) **Ground-Truth-Verb**, which select verbs based on top-1 prediction, top-5 predictions, and corresponding ground truth, respectively. If verb predictions are incorrect in the Top-1-verb and Top-5-verb settings, the corresponding semantic role predictions are also considered false.

### 4.2 Comparisons with State-of-the-Arts

**Baseline models.** Existing SR models can be classified into: 1) CRF [73]: CRF-based model, 2) CRF+DataAug [72]: CRF-based model with data augmentation, 3) VGG+RNN [42]: RNN-based prediction model with VGG backbone, 4) FC-Graph [36]: GNN-based model with fully connected semantic roles, 5) CAQ [22]: Query-based model with top-down attention, 6) Kernel-Graph [54]: GNN-based model with mixture kernel attention. Correspondingly, existing SR models can be divided as: 1) ISL [47]: RNN-based method has independent semantic role values and grounding predictions. 2) JSL [47]: RNN-based methods jointly predict semantic role values and their basis. 3) GSRTR [19]: Transformer two-stage model has both
Table 2: Performance (%) comparisons of GSRFormer (ours) and baseline methods on SWiG dataset test set.

| Models                  | Top-1-Verb | Ground-Truth-Verb |
|-------------------------|------------|-------------------|
| verb value val-all grnd grnd-all | Top-5-Verb | verb value val-all grnd grnd-all |
| ISL [47]                | 39.36      | 73.42             |
| JSL [47]                | 39.36      | 73.42             |
| GSRTR [19]              | 40.63      | 74.11             |
| SituFormer [65]         | 44.20      | 74.11             |
| CoFormer [18]           | 44.20      | 74.11             |
| GSRFormer (ours)        | 46.53      | 74.11             |

Table 3: Effectiveness of each component of GSRFormer.

| Components               | verb value val-all grnd grnd-all | verb value val-all grnd grnd-all |
|--------------------------|----------------------------------|----------------------------------|
| w/o Encoder-1st          | 35.30                            | 65.51                            |
| Gains (A)                | -11.23                           | 10.60                            |
| w/o Encoder-2nd          | 32.84                            | 79.81                            |
| Gains (A)                | -13.69                           | 79.81                            |
| w/o Decoder              | 34.94                            | 79.81                            |
| Gains (A)                | -11.59                           | 79.81                            |
| w/o Iteration            | 39.10                            | 79.81                            |
| Gains (A)                | -7.43                            | 79.81                            |
| w/o Alternate            | 35.81                            | 79.81                            |
| Gains (A)                | -10.72                           | 79.81                            |
| w/o Message              | 38.06                            | 79.81                            |
| Gains (A)                | -8.47                            | 79.81                            |
| GSR-Former (ours)        | 46.53                            | 79.81                            |

Table 4: Effects of adopting two opposite refinement orders.

| Order | verb value val-all grnd grnd-all | verb value val-all grnd grnd-all |
|-------|----------------------------------|----------------------------------|
| Refine-Verb-First        | -10.72                           | 79.81                            |
| Refine-Noun-First        | -10.72                           | 79.81                            |

Table 5: Effects of utilizing different aggregate functions.

| Aggregate Functions      | verb value val-all grnd grnd-all | verb value val-all grnd grnd-all |
|--------------------------|----------------------------------|----------------------------------|
| Element-wise Sum         | 75.64                            | 42.79                            |
| Max-Pooling              | 76.30                            | 43.58                            |
| Aggregated Message [34]  | 77.79                            | 43.58                            |
| Gated Function [38]      | 78.81                            | 44.68                            |

4.3 Ablation Studies

Effectiveness of encoder: We perform ablation studies on two multi-head attention modules of the encoder (denoted as Encoder-1st and Encoder-2nd), as shown in Table 3. Experimental results show that both structures significantly improve performance (over 10% absolute improvement). This fully demonstrates the effectiveness of the encoder module. Encoder-1st is valuable in understanding visual grounding details (comparable on grnd-all metrics), while Encoder-2nd is more conducive to mining verbs and semantic roles.

Effectiveness of decoder: We validate the entire decoder module. By removing the decoder, the modified model resembles a simple two-stage approach [19]. The results in Table 3 show that the absolute improvement is also over 10%, which fully demonstrates the effectiveness of the decoder module. Below we will analyze each part of the decoder in detail.

Effectiveness of iterative refinement: We verify iterative refinement. Without iterative refinement (i.e., only 1 transformer layer in the decoder instead of 5), the performance (2%-5% drop) will be similar to traditional RNN-based methods (e.g., JSL [47]). It hints that the real improvement to the transformer structure is the ability to make iterative improvements.

Effectiveness of alternate optimization: We verify alternate optimization. By removing this, we mean to update the verb with support verbs, and update nouns with other nouns. We are surprised to find that alternate optimization resulted in more performance drop than iterative refinement, except for the visual relevant grnd-all metrics. This fully illustrates that the core of iterative refinement is to exploit the semantic relationship between verbs and nouns (roles) in both directions.

Effectiveness of message computation: We compare with that removes the message passing mechanism, i.e., directly using features of transformer encoder to learn the semantic relations. As
shown in Table 3, message passing not only effectively uses semantic relations to improve verb and noun prediction, but also learns the visual grounding information (grnd and grnd-all metrics).

**Effects of refinement order.** Table 4 shows that refining the noun first gives better results. This is the exact opposite of the previous work [19, 42, 47]. We speculate that this is because the set of noun entities is more relevant to the visual groundings, so predicting nouns first leads to fewer errors than verbs. For example, humans naturally recognize noun entities as evidence for judgments when stating verbs. This inspires us to prioritize noun entities.

**Effects of aggregate functions.** We test four aggregate functions. Table 5 shows that Gated Function [38] achieves the best results and Element-Wise Sum does not reach satisfactory results. This contrast points out that aggregation functions should strive to highlight semantically relevant features while ignoring unnecessary noise. In future work, we will explore more fusion strategies.

**Effects of the number of stacked layers.** We validate the effect of stacking layers in three multi-head attention structures. As shown in Figure 6, the best performance is achieved when six layers are stacked in the encoder and five layers in the decoder. As the number of stacked layers increases, the performance first increases and then decreases. We attribute this performance change to the noise of stacking too many layers.

**Effects of the size of support verb set.** As shown in Figure 6 (d), the best performance is when the support set size is 5. The performance drops when the size is 1, indicating that similar verbs can support semantic understanding. Moreover, when the size is 10, the performance does not further improve, which means that expanding support verbs also raises noise.

### 4.4 Visualization and Application

We visualize the results of GSRFormer in Figure 5 (a). With the help of alternate semantic refinement, GSRFormer predicts some almost indistinguishable action verbs and semantic roles (see red font). We also show an example of the semantic question-answering application in Figure 5 (b). For example, when we ask questions like “How are women’s eyebrows made?”, GSRFormer can not only utilize the generated verb-frame to understand the question, but also provide structured answers with rich image facts.

### 5 CONCLUSION

In this paper, we first reveal the problems of existing frameworks and point out that the use of semantic relations is the root of the GSR task. To this end, we propose GSRFormer, a two-stage transformer framework that utilizes bidirectional semantic relations to iteratively refine predictions of verb and noun entities. Experimental results on SWiG dataset show that it outperforms other state-of-the-art methods. In the future, we will further explore to explain the semantic structure of GSRFormer and extend it to other semantic analysis tasks.

### ACKNOWLEDGMENTS

This work was supported by the Air Force Research Laboratory under agreement number FA8750-19-2-0200; the Defense Advanced Research Projects Agency (DARPA) grants funded under the GAILA program (award HR00111990063); the Defense Advanced Research Projects Agency (DARPA) grants funded under the AIDA program (FA8750-18-20018).

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.
A IMPLEMENTATION DETAILS

Network structure. We use ImageNet pre-trained ResNet-50 as the backbone. To facilitate computation, we use 1×1 convolution to compress the dimension of image features to 512, which is consistent with the hidden dimension of each semantic role query and verb token. To further speed up inference, the dimension of verb embedding and role embedding is 256. Correspondingly, we use learnable 2D embeddings for positional encodings, which have the same dimensions as sequences of visual and verb tokens. The number of heads for all multi-head attention blocks is 8. The sizes of the hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively. The hidden dimensions of the four structures are 2048, 1024, 1024, and 512, respectively. The bounding box regressor is three fully connected layers for target verb and noun labels with label smoothing factors of 0.1 and 0.05, respectively.
How do people get treated? What treatment do people use and where do they treat it?

How do people cook? What do people cook with, and where do people cook?

How do people maintain their house? What do people use to maintain their houses?

Figure 7: Comparison between GSRFormer and GSRTR [19]. The first two columns compare the verbs detected by GSRTR and GSRFormer, respectively. The last two columns are the semantic roles predicted by GSRFormer. With the help of an iterative refinement mechanism, GSRFormer predicts more correct verbs and nouns (marked in red font) under the Top-5-Verb setting. These examples also clearly illustrate the importance of semantic relations for GSR tasks, i.e., bidirectional semantic ties can mutually refine the predictions of verbs and nouns.

Figure 8: Application example. GSRFormer can serve cross-modal semantic question answering and reasoning on the basis of human event understanding. For example, when questioned "How do people cook? What do people cook with, and where do people cook?", GSRFormer can list cooking procedures and steps. Compared to image captioning and scene graphs, GSRFormer can not only utilize the generated structured verb-frame to apprehend the questions, but also provide answers with intuitive image facts to help users understand.