Neural Event Semantics for Grounded Language Understanding

Shyamal Buch **Li Fei-Fei** **Noah D. Goodman**

Stanford University, United States
{shyamal,feifeili}@cs.stanford.edu ngoodman@stanford.edu

Abstract

We present a new conjunctivist framework, neural event semantics (NES), for compositional grounded language understanding. Our approach treats all words as classifiers that compose to form a sentence meaning by multiplying output scores. These classifiers apply to spatial regions (events) and NES derives its semantic structure from language by routing events to different classifier argument inputs via soft attention. NES is trainable end-to-end by gradient descent with minimal supervision. We evaluate our method on compositional grounded language tasks in controlled synthetic and real-world settings. NES offers stronger generalization capability than standard function-based compositional frameworks, while improving accuracy over state-of-the-art neural methods on real-world language tasks.

1 Introduction

Capturing the compositional semantics of grounded language is a long-standing goal in natural language processing. Composition yields systematicity, and is thus essential to developing systems that can generalize broadly in real-world settings. Recent progress with neural module networks (Andreas et al., 2016b; Hu et al., 2017) and related models (Johnson et al., 2017b; Yi et al., 2018; Bahdanau et al., 2019a) have moved neural network methods closer to this goal.

These works are largely based on the idea, *functionism* (Montague, 1970), that semantic composition is function composition. In Figure 1(a), function predicates compose by nesting: Predicates like ‘‘red’’ and ‘‘circle’’ operate on sets of elements, progressively filtering them at each step (`circle(red(x))`). The final relational predicate `above` is thus several steps removed from the original inputs `x,y`. Similarly, in module networks, atomic module blocks compose by sequentially passing outputs of intermediate blocks to later modules. The diverse composition ruleset needed to coordinate function inputs and outputs leads to complexity in this paradigm, which has practical implications for its fundamental learnability. Indeed, neural module network instantiations of this framework often depend on low-level ground truth module layout programs (Johnson et al., 2017b) or large amounts of training data to sustain end-to-end reinforcement learning methods (Yi et al., 2018; Mao et al., 2019).

While functionism is the dominant paradigm in linguistic semantics, there is an intriguing alternative: event semantics (Davidson, 1967). *Conjunctivism* (Pietroski, 2005) is a particularly powerful version of event semantics, wherein the only composition operator is conjunction—structure arises by *routing* event variables to the function predicates. We illustrate this key difference between paradigms in Figure 1: in a conjunctivist setting (Figure 1(b)), even the relational `above` has events e_1, e_2 *directly routed* as input, rather than taking inputs that are output results of a sequence of `filter` operations. Overall meaning is still preserved, since e_1 *concurrently routes* to (`red`, `circle`) and e_2 to (`green`, `square`). All module outputs directly contribute to the final truth value without intermediate steps. Altogether, this shift from deriving compositional structure by functional module layout to conjunctive events routing offers a path to improved learnability; we explore the implications of this line of thinking in the context of compositional neural models.

We propose neural event semantics (NES), a new conjunctivist framework for compositional grounded language understanding. Our work addresses the drawbacks of modern neural module approaches by re-examining the underlying

Project Website: https://neural-event-semantics.github.io/.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 875–890, 2021. https://doi.org/10.1162/tacl_a_00402
Action Editor: Luke Zettlemoyer. Submission batch: 3/2021; Revision batch: 4/2021; Published 8/2021.

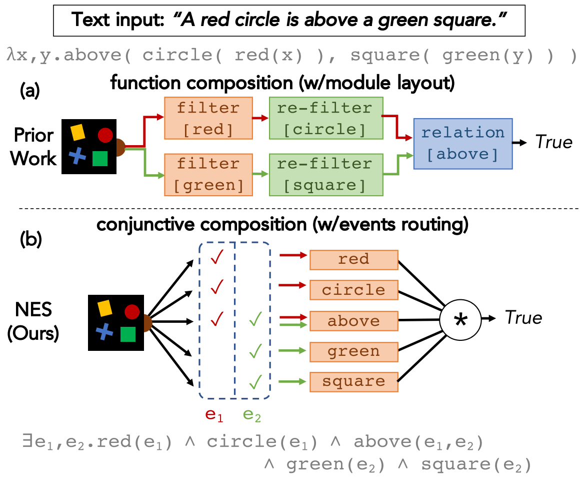

Text input: "A red circle is above a green square."
λx,y.above(circle(red(x)), square(green(y)))
(a)
function composition (w/module layout)
Prior Work
filter [red]
re-filter [circle]
filter [green]
re-filter [square]
relation [above]
True
(b)
conjunctive composition (w/events routing)
NES (Ours)
✓
✓
✓
✓
✓
✓
red
circle
above
green
square
*
True
e1 e2
∃e1,e2.red(e1) ∧ circle(e1) ∧ above(e1,e2) ∧ green(e2) ∧ square(e2)

Figure 1: **(a)** Prior neural network methods for compositional semantics, such as neural module networks, derive compositional structure through nested application of function modules. This paradigm, rooted in *functionism*, is powerful but retains drawbacks to learnability due to its complexity. **(b)** We propose neural event semantics (NES), a new framework based on *conjunctivism*, where all words are classifiers and output scores compose by simple multiplication. We call the input spatial regions to these classifiers *events*: NES derives semantic structure from language by learning how to *route event inputs* of classifiers for different words in a context-sensitive manner. By relaxing this routing operation with soft attention, NES enables end-to-end differentiable learning without low-level supervision for compositional grounded language understanding.

semantics framework, shifting from functionism to conjunctivism. The focus of NES revolves around *event* variables, abstractions of entities in the world (e.g., in images, we can think of events as spatial regions). We treat *all words as event classifiers*: For each word, a single score indicates the presence of a concept on a specific input (e.g., `red`, `above` in Figure 1(b)). We *compose output scores* from classifiers by *multiplication*, generalizing logical conjunction. The structural heart of NES is the intermediate *soft (attentional) event routing* stage, which ensures that these otherwise independent word-level modules receive contextually consistent event inputs. In this way, the simple product of all classifier scores accurately represents the intended compositional structure of the full sentence. Our NES framework is end-to-end differentiable, able to learn from high-level supervision by gradient descent while providing interpretability at the level of individual words.

We evaluate our NES framework on a series of grounded language tasks aimed at assessing its generalizability. We verify the merits of our conjunctivist design in a controlled comparison with functionist methods on the synthetic ShapeWorld benchmark (Kuhnle and Copestake, 2017). We show NES exhibits stronger systematic generalization over prior techniques, without requiring any low-level supervision. Further, we verify the flexibility of the framework in real-world language settings, offering significant gains (+4 to 6 points) in the state-of-the-art accuracy for language and zero-shot generalization tasks on the CiC reference game benchmark (Achlioptas et al., 2019).

2 Background and Related Work

Compositional Neural Networks. The advent of neural module networks (NMN) (Andreas et al., 2016a,b; Hu et al., 2017) and related techniques (Johnson et al., 2017b; Yi et al., 2018; Bahdanau et al., 2019a) has proven to be a driving force in compositional language understanding. These techniques share a key principle: Small, reusable neural network modules stack together as functional building blocks in an overall executable neural program. A parsing system determines the programmatic layout, wiring the outputs of intermediate blocks to the inputs of other blocks.

The reliance of these techniques on prespecified module libraries, ground truth supervision on functional module layouts, and/or sample-inefficient reinforcement learning methods (Williams, 1992) has motivated subsequent work to eschew explicit semantics for recurrent attentional computation techniques (Hudson and Manning, 2018; Perez et al., 2018; Hu et al., 2018; Hudson and Manning, 2019). This class of more *implicit* semantics methods offers the benefits of end-to-end differentiability of traditional non-compositional neural networks (Lake et al., 2017), making them better suited for real-world settings. As a trade-off, however, these methods exhibit less systematic generalization than their more explicit counterparts (Marois et al., 2018; Jayram et al., 2019; Bahdanau et al., 2019b).

Recent work has also suggested that the modular network approach leads to limitations of systematic generalization: Functional module layout can lead to entangled concept understanding (Bahdanau et al., 2019a; Subramanian et al., 2020). While these works go on to propose mitigating measures, such as module-level pretraining, we consider an orthogonal approach: re-visiting the underlying semantics foundation. This enables us to address the challenges jointly: Our NES framework retains the end-to-end learnability of implicit methods, while improving upon the systematic generalizability of explicit ones.

Grounded Compositional Semantics. Our work is also closely related to the broader, pre-neural network body of prior work which developed models for compositional semantics in grounded language settings (Matuszek et al., 2012; Krishnamurthy and Kollar 2013; 2014). These methods all share the two-stage approach of semantic parsing and evaluation, and combine functionist *and* conjunctivist elements. The parsing stage typically leverages a (functionist) combinatory categorial grammar (CCG) parser (Zettlemoyer and Collins, 2005) to map input language input to a discrete (conjunctive) logical form bound by an existential closure. The evaluation stage passes visual segments as input to these predicates to obtain a final score representing its truth condition. In our work, we aim to generalize these frameworks to a modular neural network setting, embracing conjunctivist design across all stages to improve end-to-end learnability. Our proposed soft event routing mechanism relaxes prior discrete constraints and offers an alternative to probablistic program (Krishnamurthy et al., 2016) formulations. Together, NES is able to learn how to predict the (soft) conjunctive neural logical forms while jointly learning the underlying semantics of each concept (without pre-specification) end-to-end from denotation alone.

Grounded Language Understanding. The space of grounded language understanding methods and tasks is large, encompassing tasks in image-caption agreement (Kuhnle and Copestake, 2017; Suhr et al., 2019), reference grounding (Monroe et al., 2017; Achlioptas et al., 2019), instruction following (Ruis et al., 2020; Vogel and Jurafsky, 2010; Chaplot et al., 2018), captioning (Chen et al., 2015), and question answering (Antol et al., 2015; Johnson et al., 2017a; Hudson and Manning, 2019), among others. Often, the ability to operate with only high-level labels is critical (Karpathy and Fei-Fei, 2015). Consistent with recent work (Bahdanau et al., 2019a), we center our analysis on foundational tasks of caption agreement and reference grounding, on both synthetic and real-world language data, with the understanding that core insights can translate to related tasks.

3 Technical Approach

3.1 Prelude: Classical Conjunctivism to NES

To explain our proposed differentiable neural approach, we first revisit classical logic in our current context. In conjunctivist event semantics (Pietroski, 2005), we work with the space of existentially quantified conjunctions of predicates. For illustration, consider the partial logical form:

$$\exists e_1, e_2 \in V. \, [[\text{circle}(e_1) \wedge \text{on}(e_1, e_2)]] \quad (1)$$

where e_i are event variables and V is the domain of candidate event values. To evaluate this expression, we need an *interpretation* of the variables: an assignment of event values in V to each event variable e_i. We then *route* these events to the arguments of predicates based on the logical form. The logical form gives the abstract template for which events route to which inputs and, most crucially, which arguments are shared across predicates (e_1 routes to ''circle'' and the first argument of ''on''). We make this routing explicit by a **routing tensor** $A_{wri} \in \{0,1\}$: For each argument slot (r) of each predicate (w, for word), $A_{wr*} \in \{0,1\}^n$ is a one-hot vector indicating *which* of the n event variables $e_i \in \mathbf{e}$ belongs in this argument slot. We can thus rewrite the matrix expression in Equation (1) as:

$$[[\text{circle}(A_{11*}\mathbf{e}, A_{12*}\mathbf{e}) \wedge \text{on}(A_{21*}\mathbf{e}, A_{22*}\mathbf{e})]] \quad (2)$$

Without loss of generality,[1] we upgrade each predicate to take a fixed m arguments; here $m = 2$. Equation (2) makes it clear that the routing tensor A is the *key syntactical element specifying the structure of the logical form* in Equation (1). Having routed events e_i to predicate arguments via A, we can evaluate the predicates (''circle'', ''on''). These predicates are Boolean functions, assigned by a lookup table (*lexicon*). We compose the outputs of these Boolean functions by conjunction to get the truth-value of the entire matrix. This describes how we evaluate the matrix expression in Equation (1) for a *specific* assignment of e_i in V. We arrive at the final interpretation of

[1] We add a ***background event variable*** $e_\emptyset$, backed by a null representation; A can route $e_\emptyset$ to extra slots. In Equation (2), routed events to ''circle'' are $A_{11*}\mathbf{e} = e_1$ and $A_{12*}\mathbf{e} = e_\emptyset$.

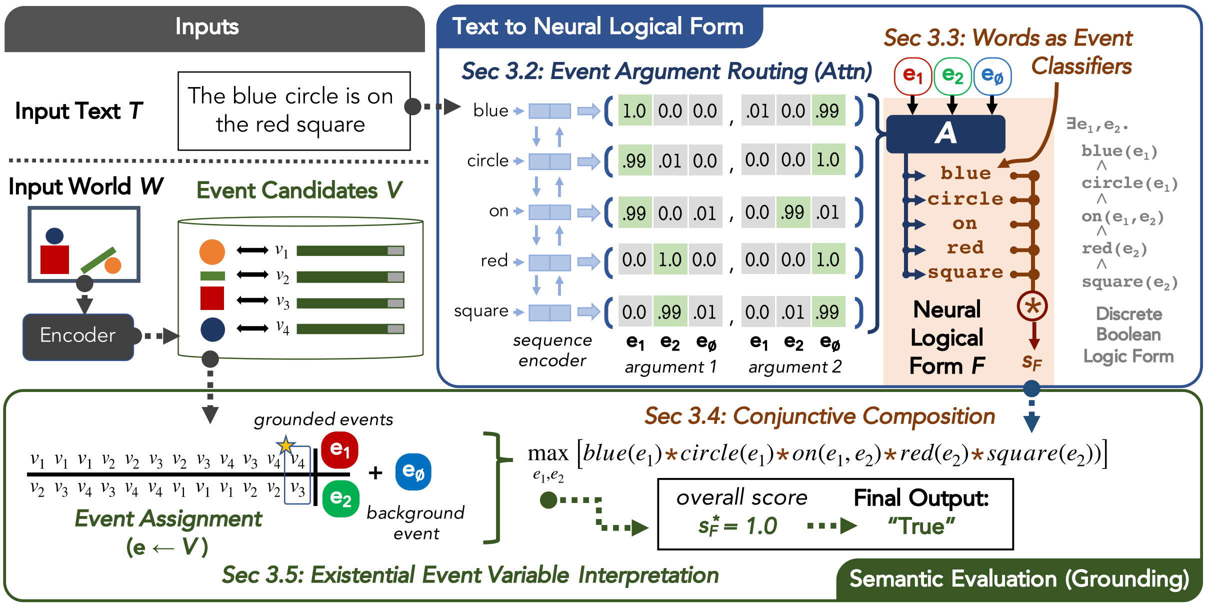

Inputs
Input Text T
The blue circle is on the red square
Input World W
Event Candidates V
v_1
v_2
v_3
v_4
Encoder
Text to Neural Logical Form
Sec 3.2: Event Argument Routing (Attn)
blue
circle
on
red
square
1.0 0.0 0.0 , .01 0.0 .99
.99 .01 0.0 , 0.0 0.0 1.0
.99 0.0 .01 , 0.0 .99 .01
0.0 1.0 0.0 , 0.0 0.0 1.0
0.0 .99 .01 , 0.0 .01 .99
sequence encoder
e_1 e_2 $e_\emptyset$
argument 1
e_1 e_2 $e_\emptyset$
argument 2
Sec 3.3: Words as Event Classifiers
e_1 e_2 $e_\emptyset$
A
blue
circle
on
red
square
Neural Logical Form F
s_F
$\exists e_1, e_2.$ blue(e_1) ∧ circle(e_1) ∧ on(e_1,e_2) ∧ red(e_2) ∧ square(e_2)
Discrete Boolean Logic Form
grounded events
v_1 v_1 v_1 v_2 v_2 v_3 v_2 v_3 v_4 v_3 v_4 v_4
v_2 v_3 v_4 v_3 v_4 v_4 v_1 v_1 v_1 v_2 v_2 v_3
e_1
e_2
+ $e_\emptyset$
background event
Event Assignment ($e \leftarrow V$)
Sec 3.4: Conjunctive Composition
$\max_{e_1,e_2} [blue(e_1) \ast circle(e_1) \ast on(e_1,e_2) \ast red(e_2) \ast square(e_2))]$
overall score $s_F^\ast = 1.0$
Final Output: "True"
Sec 3.5: Existential Event Variable Interpretation
Semantic Evaluation (Grounding)

Figure 2: We propose **neural event semantics (NES)**, an end-to-end differentiable framework based on conjunctivist event semantics (Sec 3.1). NES parses input text to a neural logical form F, which can score a given set of input events. In NES, all words are event classifiers (Section 3.3) whose scores compose by multiplication (Section 3.4). The structural heart of NES is a differentiable event argument routing operation (Section 3.2), ensuring arguments to each event classifier are contextually correct. NES semantically grounds F to an input world W by existential event variable intepretation (Section 3.5), finding a satisfying assignment (if one exists) of events $\mathbf{e}$ from values V.

Equation (1) by *existential quantification*: searching over the possible assignments to see if there exists one that makes the matrix true.

We emphasize that the logical form is *fully determined* by the routing tensor A and the lexicon mapping each word/predicate to a Boolean function. Evaluation is specified by conjunctive composition and finding a satisfying variable interpretation. Our strategy to develop a learnable framework is to *soften* each of the key components: **argument routing** (Section 3.2), **predicate evaluation** (Section 3.3), **conjunctive composition** (Section 3.4), and **existential event interpretation** (Section 3.5).

Overview. We propose a **neural event semantics (NES)** framework, illustrated in Figure 2, which relaxes this classical logic into a differentiable computation that can be learned end-to-end. NES takes a text statement T and constructs a neural logical form F. This form is specified by a now *real-valued* routing tensor $A_{wri} \in [0, 1]$ and a lexicon associating event classifiers M_w to each word w. NES specifies composition via the product of classifier prediction scores, as a relaxation of conjunction. Finally, evaluation is completed by existentially interpreting event variables e_i into a domain of event values V (grounded representations extracted from a visual world W) by a max operator.

3.2 Differentiable Event Argument Routing

Our first key operation in NES is to predict the argument routing tensor A from the input language. Critically, we relax A from its original discrete formulation in Section 3.1 to a continuous-valued one $A_{wri} \in [0, 1]$, where $A_{wr\ast} \in [0, 1]^n$ is normalized by softmax over the index for the n events e_i. This softened *routing* can be seen as a form of *attention*, determining which argument slot r for a word w will attend to which event variables e_i (see Figure 3). We predict these attentions directly from the input tokenized text sequence $T = [t_1, \ldots, t_l]$, of length l. For each token word t_w, we pass a word embedding q_w as input to a bidirectional LSTM (Graves and Schmidhuber, 2005) that serves as the *sequence encoder* and outputs forward/backward hidden states $(h_w^{\rightarrow}, h_w^{\leftarrow} \in \mathbb{R}^{d_h})$ capturing the bidirectional context surrounding t_w. Passing the *concatenated* states through a linear layer, we obtain a final hidden state:

$$h_w = W([h_w^{\rightarrow}, h_w^{\leftarrow}]) + b \quad \in \mathbb{R}^{d_h} \qquad (3)$$

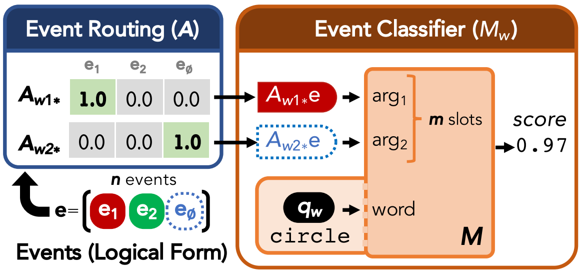


Figure 3: **Words as Classifiers of Routed Events.** All words w correspond to modules M_w of a single type signature. Predicted argument routing attention A routes input events e from the *overall* logical form F to the *specific* arguments in the event classifier M_w (per Equation (5)), ensuring contextual consistency between event classifiers for different words. q_w, a *decontextualized* word embedding, indicates to M_w its lexical concept. M_w shown here with maximum arity $m = 2$ slots and $n = 3$ events (including the ungrounded background event $e_\emptyset$); since ''circle'' only binds to one argument e_1, the second slot is bound to $e_\emptyset$. See Section 3.2 and 3.3.

From h_w, a multilayer perceptron (MLP$_{\text{ROUTE}}$) network outputs for each argument slot r:

$$A_{wr*} = \text{softmax}(\text{MLP}_{\text{ROUTE}}(h_w)) \qquad (4)$$

Over the full input sequence of length l, we obtain the full argument routing tensor $A \in [0,1]^{l \times m \times n}$, with m argument slots per word and n events. Note that the prediction of A from input text T plays the role of capturing *syntax* for NES, using the language to derive coordination of argument routings across different words.[2]

A key design aspect of the routing operation: Because A can route an ungrounded *background event* $e_\emptyset$ to (extra) argument slots, NES can implicitly learn the *arity* of each word. Further, the attention formulation enables *partial* routing of such background events; we observe later in Section 4.1.4 that this is critical to enabling the more complex coordination necessary to handle negation.

3.3 Words as Event Classifiers

In NES, all words are event classifiers: Words are associated with modules M_w that output a real-valued score s_w of how true a lexical concept is for a given set of (routed) event inputs (Section 3.2, Figure 3). Denoting events $e_i \in \mathbb{R}^{d_e}$, $e_\emptyset$ as a null background event, and $\mathbf{e} = [e_1 \cdots e_{n-1} \mid e_\emptyset] \in \mathbb{R}^{n \times d_e}$, we can formalize the routed inputs as $A_{wr*}\mathbf{e} \in \mathbb{R}^{d_e}$. The concatenation of these routed inputs over all m argument slots is input to M_w.

While in principle the modules can be completely separate for each word in the *lexicon*, we choose to share the weights of the different classifiers M_w: This improves memory efficiency for large vocabularies and is helpful in real-world language generalization settings. Thus, we can realize modules M_w by an MLP network that receives the word embedding q_w as further input (see Figure 3), computing its output s_w as:

$$s_w = \sigma(\text{MLP}_{M_w}([A_{w1*}\mathbf{e}, \ldots, A_{wm*}\mathbf{e}; q_w])) \qquad (5)$$

where σ denotes the sigmoid function that normalizes the output score $s_w \in [0,1]$.[3]

3.4 Conjunctive Composition in NES

Per Section 3.1, the matrix of a classical conjunctive logical form (for a given interpretation of variables) is evaluated by composing Boolean predicate outputs by *conjunction*. For the neural logical form F in NES, we consider the real-valued generalization of conjunction: We compose the l word-level scores s_w from the classifiers M_w (Equation (5)) by *multiplication* ($\prod_w s_w$). For numerical stability, we calculate the combined log score in log space:

$$\log s_F = \frac{1}{l} \sum_w \log s_w \qquad (6)$$

where the length normalization is optional but helps with training on variable length sequences.

3.5 Existential Event Variable Interpretation

In previous Section 3.2-3.4, we've described how NES translates input language to a neural logical form F, and how such a logical form can operate for a *specific* intepretation (binding) of events to candidate values V. Now, we describe the final existential variable interpretation step, which relaxes the existential quantification of classical logic (Equation (1)) into a max operation

[2] We emphasize that this is a language-only operation: Coordination here is *not* conditional on the later grounding step to specific event values V in the visual world (Section 3.5).

[3] q_w is a *decontextualized* embedding that only represents the standalone lexical concept, *not* the recurrent embedding h_w. Consistent with Subramanian et al. (2020), we find this improves systematic generalization in NES and baselines.

over possible event interpretations of a specific input domain V.

Candidate Event Values V. We decompose our input world W into a set of candidate event proposals, with corresponding representation values V. In our experiments, we process input visual scenes W with a pre-trained convolutional visual encoder ϕ (Simonyan and Zisserman, 2015; He et al., 2016) to provide a set of up to k candidate event value representations $V = \{v_1, \ldots, v_k\}$, with $v \in \mathbb{R}^{d_e}$. These candidate values capture the information corresponding to the localized image segment surrounding that specific event; we base our approach on recent findings of object-centric representations for compositional modular network approaches (Yi et al., 2018). To capture spatial information, we augment each representation with the spatial coordinates of the center of its bounding box; this enables NES and our relevant baseline methods (e.g., NMN) to assess the semantics of spatial relationships (e.g., ''below'') while operating directly on event values.

Assignment and Final Scoring. Given the domain V of candidate event values, an interpretation is thus an assignment of each of the $n - 1$ grounded event variables (we don't include $e_\emptyset$) to a unique value in V: We denote this assignment operation as $\mathbf{e} \leftarrow V$. We translate the existential closure ($\exists e_1, e_2$ in Figure 2) as an operation that determines the best scoring assignment of event candidate values to event variables. Expanded, the final grounded score $s^*_F = \max_{\mathbf{e} \leftarrow V} s_F$ is:

$$s^*_F = \max_{\mathbf{e} \leftarrow V} \frac{1}{l} \sum_{w} \log M_w(A_{w1*}\mathbf{e}, \ldots, A_{wm*}\mathbf{e}; q_w) \tag{7}$$

Figure 4 visualizes output score tables (including s_w, s_F, s^*_F) with $k = 2$ candidate event values and $n = 3$ events including background $e_\emptyset$. We highlight that Figure 4 shows how each individual module provides consistent outputs depending on the specific event interpretation $\mathbf{e} \leftarrow V$ (e.g., ''below'' is only true if (e_1, e_2) bind to (v_2, v_1), not (v_1, v_2)). The final score s^*_F reflects the s_F of that correct assignment, since it is the max score.

3.6 Training: Learning from Denotation

We train our overall system end-to-end with gradient descent with a dataset of (statement T, world scene W, true/false denotation label Y) triplets.

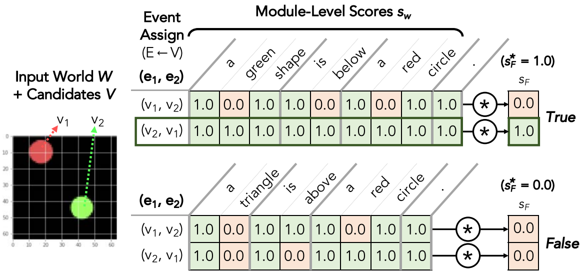


Figure 4: **Qualitative Results (Scoring).** Example end-to-end NES results on ShapeWorld. We show an input world with two event candidates (for clarity) with representations v_1, v_2 for the red and green circles, respectively. We visualize the possible event assignments $(e_1, e_2) \in \{(v_1, v_2), (v_2, v_1)\}$ and the classifier scores $s_w \in [0, 1]$ for each assignment, including stop words. We find NES provides correct and consistent predictions across assignments and concepts, *without* any explicit logical form-level supervision. See Section 4.1.4.

We apply a straightforward binary cross entropy loss at the level of text statements and their truth labels to the final output score s^*_F, *without needing any low-level ground truth supervision* of the neural logical form. Overall, our full NES framework offers advantages from both traditional neural module network methods and end-to-end differentiable implicit semantics techniques.

The max operation in Equation (7) is a technical challenge for the end-to-end *training*. To improve gradient flow, we propose to use a tunable approximation $f_{\max}$, which approaches the max as $\beta \rightarrow \infty$ and is always *upper*-bounded by it:

$$f_{\max}(\mathbf{s}; \beta) = \frac{\sum_q (\mathbf{s}_q)^{\beta+1}}{\sum_q (\mathbf{s}_q)^{\beta}} \leq \max(\mathbf{s}) \tag{8}$$

In context, $\mathbf{s}$ is a vector of all the scores s_F (Equation (6)) corresponding to the assignments $\mathbf{e} \leftarrow V$, and the output of Equation (8) is a bounded approximation of s^*_F in Equation (7). See Appendix A for correctness and details.[4] During *test-time inference*, we still use the original max operation shown in Equation (7).

[4] Bound follows from Hölder's inequality. Equation 8 is important since standard alternatives (e.g., log-sum-exp) do not have this upper-bound, and the possibility of multiple valid assignments $\mathbf{e} \leftarrow V$ renders softmax inappropriate. Since f_{max} converges quickly to the max operation as β increases, we can use numerically stable values $\beta \leq 4$ during training.

4 Experiments

4.1 Experiments: Synthetic Language

We design the first series of experiments to highlight key compositional and generalization properties of NES in a controlled, synthetic setting.

4.1.1 Dataset and Tasks

ShapeWorld. Our synthetic tasks and datasets are based on the ShapeWorld benchmark suite (Kuhnle and Copestake, 2017), which was designed specifically for evaluation of compositional models for grounded semantics. Here, events are based on simple objects: shapes with different color attributes and spatial relationships. Images are generated by sampling events from task-specific distributions with visual noise (e.g., hue, size variance), and are placed without hard grid constraints. For each image, multiple true/false language statements are generated with a templated grammar (Copestake et al., 2016). Negative statements are generated close to the distribution of positive statements to ensure difficulty: Models must understand *all* aspects of the statement correctly to output a truth condition label. We visualize an example in the qualitative results (Figure 4).

Task A: Standard Generalization. This generalization task evaluates compositional models on the standard setting where train and evaluation splits are based on the same underlying input event distribution. This task is similar to the original SHAPES dataset (Andreas et al., 2016b), without shape positions locked to a 3×3 spatial grid.

Task B: Compositional Generalization. The compositional generalization task examines the systematic generalization of models to an *unseen* event distribution. During test time, every instance has at least one event sampled from a held-out distribution. For example, while *red* triangles and *blue* squares may be present at train time, *blue* triangles and *red* squares are only present during test time. Critically, any *language* associated with these unseen events is *always false* during training since these events are never actually present. Thus, models that overfit on complete phrases during training will not generalize well at test time.

Task Variant: Negation. For both tasks, we include a variation with negation to ensure NES can model non-intersective modifiers, which are prevalent in real-world grounded language. In these variants, true and false statements that include attribute-level negation (e.g., phrases like ‘‘not red’’) are also generated for each image.

4.1.2 Baseline and Model Details

Baselines. Across our synthetic experiments, we compare NES against baselines in 3 categories:

- **Black-box neural networks**. These baseline neural network models combine CNN, LSTM, and attention components (Johnson et al., 2017a) and represent standard end-to-end black-box techniques for language + vision tasks.

- **Functionist approaches.** For our functionist baselines, we consider the prevailing parameterizations of the neural module networks (NMN) framework (Andreas et al., 2016b). For the modules, we leverage the base generic module design introduced in the E2ENMN framework (Hu et al., 2017; Bahdanau et al., 2019a). Because our experiments are event-centric, the inputs and implementation of the framework are consistent with prior work (Yi et al., 2018; Mao et al., 2019; Subramanian et al., 2020). Thus, each module takes as input a set of localized event values (originally from the image), an attention over these values (from a preceding module step), and a decontextualized word embedding. The module then applies the attention and processes the input, before outputting an updated attention to be used in dependent downstream module steps. For end-to-end (E2E) experiments, ground truth programs are used to pre-train the parsing module layout generator, which is the structural heart of NMN. This parser is implemented using a sequence-to-sequence Bi-LSTM (Hu et al., 2017; Johnson et al., 2017b). We emphasize that, in our experiments, we ensure consistent hidden state sizes for both the modules and the sequence encoder for NMN and NES, as well as consistent event-centric visual + decontextualized word embedding input.

- **Implicit semantics methods.** This class of models leverages recursive computation units with attention over visual and textual input to provide better compositionality than traditional end-to-end black-box neural network methods. We examine the MAC model (Hudson and Manning, 2018, 2019) as a representative baseline, following recent prior work (Bahdanau et al., 2019a). Similar to our NMN baseline, we report results with an event-centric version of the MAC model, following Mao et al. (2019), such that MAC is able to attend over a discrete set of localized event values. Thus, we can enable fair and consistent comparison of MAC, NMN, and other baselines with NES.

Implementation Details. Models and baselines are implemented in PyTorch (Paszke et al., 2019). Localized event candidate values V are extracted by a pre-processing step. Our encoder ϕ is a ResNet-101 network (He et al., 2016), and localized event feature representations are based on *conv4* features per prior work (Johnson et al., 2017a; Hudson and Manning, 2018) with pixel grid coordinates (per Section 3.5) to capture the necessary spatial and visual information for the downstream semantics. Following standard work in object detection (He et al., 2017), we use pooling to ensure all localized event values have the same dimension. Word embeddings are 300-dim GloVe.6B embeddings (Pennington et al., 2014). All text and visual inputs are *consistent* across all models for fair comparison. As noted previously, model sizes are also kept consistent across models where applicable. Please refer to the supplement for implementation and additional details.[5]

4.1.3 Validating Conjunctivism

Overview. Our first experiments are centered around validating a fundamental design principle underlying our NES framework: that concept meaning *can* be effectively represented by conjunction of event classifiers. Both NMN and NES leverage syntax to guide their compositional structure: functional module layout (NMN) and event routing (NES), respectively.[6] Here, we *isolate* the impact of the design philosophy on the quality of the learned semantics by providing ground truth (GT) ''syntax'' (layout or routing) to each framework, assessing performance on Tasks A and B.

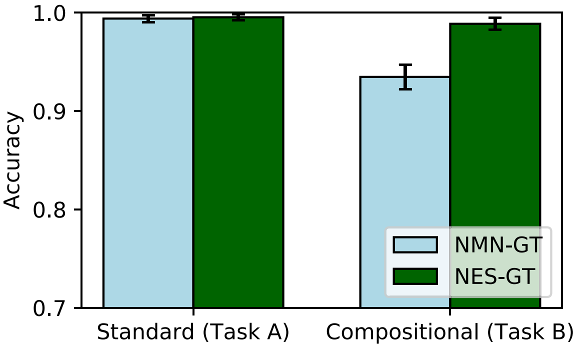


Figure 5: **Validating Conjunctivism.** Here, we provide ground truth (GT) logical forms for both functionist (NMN) and conjunctivist (NES) approaches. Controlling other factors, we observe that our conjunctivist NES framework provides better systematic generalization (Task B) than a functionist one. See Section 4.1.3.

Systematic Generalization. Figure 5 shows the results for both NMN-GT and NES-GT. Both frameworks perform equally well on the standard generalization task (Task A), showing that the NES conjunctivist design preserves the efficacy of the functionist paradigm. In Task B however, while both frameworks perform reasonably well, NES exhibits stronger systematic generalization capability than the NMN model when evaluated on an unseen event distribution. These quantitative results suggest that NES enables a stronger decoupling of individual concepts, yielding higher accuracy when they are composed for unseen events.

To explore concept disentanglement further, we analyze the color sensitivity of color words in Figure 6. For this analysis, we take the trained models from Task B and examine the normalized response score of different modules (e.g., `red`) to a continuous spectrum of color input. We sample the input shapes for each color classifier from the unseen event distribution. Our analysis suggests that NES offers stronger disentanglement of attribute concepts: color words respond to separated and appropriate spectral regions, in contrast to NMN.[7]

[5] Available at `https://neural-event-semantics.github.io/`.

[6] We note that while we focus on the functionist realizations of NMNs prevalent in prior work, we recognize that the broader family of modular network approaches can include conjunctivist elements as well. A key intention of these experiments is to illustrate the value of our conjunctivist design as a compelling direction for future modular network design.

[7] This finding, with respect to NMN, is analogous and consistent with concurrent prior work (Subramanian et al., 2020).

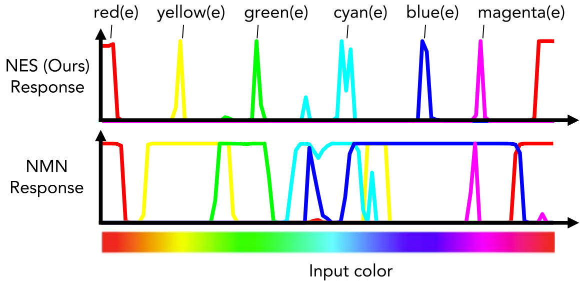


Figure 6: **Validating Conjunctivism: Attribute Response**. Response graphs for color attributes on ShapeWorld data in Task B. Our conjunctivist NES framework offers stronger disentangled understanding of each word as a concept classifier, compared to the prior functionist NMN framework. See Section 4.1.3.

4.1.4 End-to-End Experiments

Overview. Having validated that conjunctivist composition can support strong performance with known event routings, our second set of synthetic experiments are designed to assess the full end-to-end learning capability of the NES framework, including the critical event routing stage. In this setting, we offer no ground truth logical form input or supervision to the NES model, and evaluate performance on all tasks. We do necessary program layout pre-training for the E2E-Func (NMN) baseline prior to end-to-end REINFORCE training.

Generalization. In Figure 7, we show that our initial findings in Section 4.1.3 hold in the more general end-to-end setting, across the broader set of model classes. While compositional methods consistently outperform the noncompositional baselines, there is a clear differentiation between MAC and NES/E2E-Func on Task B (systematic novel-event generalization). This suggests that MAC relies too strongly on correlative associations of text phrases for unseen events, overfitting at training.

In Figure 4, we visualize a table of NES score predictions on a specific input V, using a two-event setting for visual clarity. An input statement is considered true if there is an assignment (grounding to V) of the events with a high overall score. Across different event assignments $\mathbf{e} \leftarrow V$, NES provides consistent and correct score outputs. Because NES considers each word as its own event classifier (with appropriate routing), it provides interpretable indicators for *which* attributes are specifically not present for each assignment.

In Figure 8, we visualize the event routing predictions from an example NES model trained end-to-end. Consistent with our observation in Figure 4, we see that the model can learn approximate routings and implicit *arity* of the different event classifiers. Though event routings are modeled as soft attention and classifier output scores are continuous, both have approached nearly discrete outputs by the end of learning, capturing the underlying logical structure of the domain.[8]

Negation. Finally, we demonstrate that NES is capable of handling non-intersective modifiers by examining its ability to model property negation. In contrast with functionist models, conjunctivist event semantics must handle negation through modification of the input event to the given predicate (Pietroski, 2005). In Figure 9, we show the results from these experiments. First, we observe that NES can maintain the same level of generalization accuracy in variants of Task A and B that contain negation. Visualizing an example model, we see that NES learns to coordinate negation through its *event routing* stage: the presence of ''not'' in the textual input can lead NES to predict a soft routing A_{w1*} that attends to a combination of *both* e_1 and the ungrounded background $e_\emptyset$ for the first argument of ''red'' (denoted as e_1' in the example). Now, when this specific ''red'' attribute classifier processes its updated event arguments, its classification behavior is reversed: a high score when the attribute is *not* present in the *original* e_1.

We compare with an *ablation* variant of NES that removes this routing flexibility: for attribute classifiers M_w, we restrict their routing attention A_{w**} to *only* consider the $n-1$ grounded events in the first argument slot (removing $e_\emptyset$ from consideration) and fix the second slot a_2 to the background $e_\emptyset$. Because individual event classifier modules only take *decontextualized* word embeddings, the event routing mechanism is the *only* way for context information to influence the classification. Thus, this ablation directly reflects the impact of the flexible event routing mechanism and its usage of the ungrounded background event to

[8] Without low-level supervision to break symmetry, it is possible for separate end-to-end training runs to learn different but equivalent routing schemes (and matching event classifiers): for example, NES can learn event classifiers M where argument slot 2 is consistently its primary slot (instead of slot 1). In such a case, we can use the jointly learned (consistently inverted) event routings to remap for visualization.

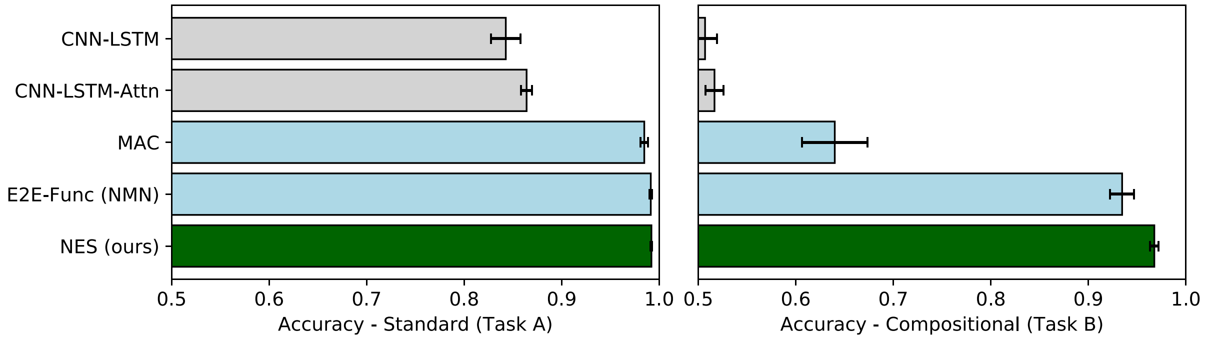


Figure 7: **End-to-End Methods.** Generalization performance of end-to-end-methods on ShapeWorld tasks. We observe that our conjunctivist NES framework offers stronger generalization performance on both standard (Task A) and systematic (Task B) compositional task settings. See Section 4.1.4 for additional details and analysis.

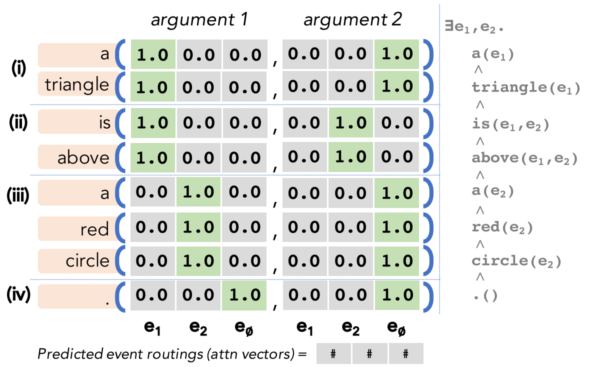

argument 1
argument 2
∃e₁,e₂.
(i)
a (1.0 0.0 0.0 , 0.0 0.0 1.0) a(e₁) ∧
triangle (1.0 0.0 0.0 , 0.0 0.0 1.0) triangle(e₁) ∧
(ii)
is (1.0 0.0 0.0 , 0.0 1.0 0.0) is(e₁,e₂) ∧
above (1.0 0.0 0.0 , 0.0 1.0 0.0) above(e₁,e₂) ∧
(iii)
a (0.0 1.0 0.0 , 0.0 0.0 1.0) a(e₂) ∧
red (0.0 1.0 0.0 , 0.0 0.0 1.0) red(e₂) ∧
circle (0.0 1.0 0.0 , 0.0 0.0 1.0) circle(e₂) ∧
(iv)
. (0.0 0.0 1.0 , 0.0 0.0 1.0) .()
e₁ e₂ e∅ e₁ e₂ e∅
Predicted event routings (attn vectors) =

Figure 8: **Learned Event Routing.** We visualize the *predicted* soft event routings of a sentence from Figure 4. **(i)** shows how ‘‘a’’ and ‘‘triangle’’ are effectively arity-1 functions, with the same event e_1 routed to their first argument, and $e_\emptyset$ to the second. **(iii)** shows the same, with e_2. **(ii)** shows an arity-2 routing for relational predicates, and **(iv)** shows how punctuation can be given an arity-0 routing. See Section 4.1.4.

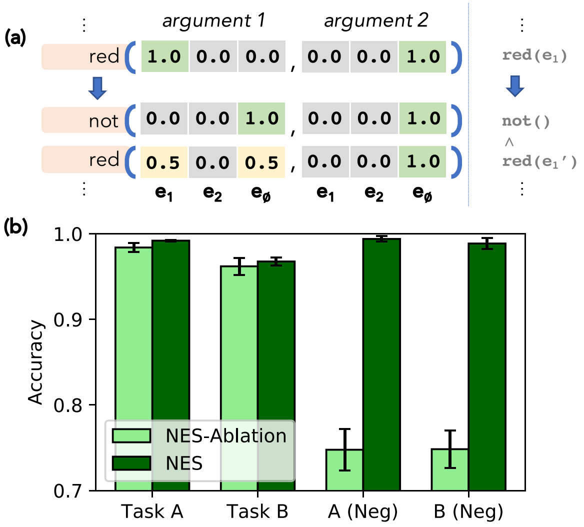


Figure 9: **Negation with NES. (a)** We visualize one way in which NES can handle coordination for non-intersective modifiers (e.g., attribute negation) by leveraging the background event $e_\emptyset$. NES soft routing leads to a modified event argument input e_1' attending over e_1 and $e_\emptyset$, enabling the `red` classifier to output the opposite prediction (now, output score = 1.0 if original e_1 is *not* red). **(b)** NES performance on Task A and B negation variants remains consistent. Ablation (Section 4.1.4) highlights the impact of the event routing mechanism.

handle more complex language settings. We find that while the ablation maintains performance on the standard tasks, its accuracy significantly decreases in this setting where some input statements have negation. Overall, we observe that the rich, augmented event space and flexible event routing stage enable our conjunctivist framework to learn how to model non-intersective modifiers, a crucial step for real-world language (Section 4.2).

4.2 Experiments: Real-World Language

Having validated the efficacy of NES in a controlled synthetic setting, we now explore NES in a grounded reference game task to demonstrate its broader applicability. Because the overall end-to-end NES framework requires no low-level supervision during training, it mirrors the broader applicability of implicit semantics methods (MAC) to less structured, human-generated language.

Chairs-in-Context (CiC). The Chairs-in-Context (CiC) dataset (Achlioptas et al., 2019) contains chairs and other objects from the ShapeNet dataset, paired with human-generated language collected in the context of a reference game. Each CiC input consists of a set of 3 chairs representing a contrastive communication context, with a human utterance (up to 33 tokens) intended to identify one of the chairs. In total, there are over 75k triplets with an 80-10-10 split for train-val-test. CiC also contains a zero-shot evaluation set with triplets of unseen object classes (e.g., tables). CiC is challenging due to its relatively long-tail language diversity and varied visual inputs.

Task A: Language Generalization. Our first CiC benchmark task is language generalization, where a model must ground the specific chair from the input set given a referring utterance. The dataset split ensures no overlap in speaker-listener pairs between training and evaluation, so models must generalize to new communication contexts.

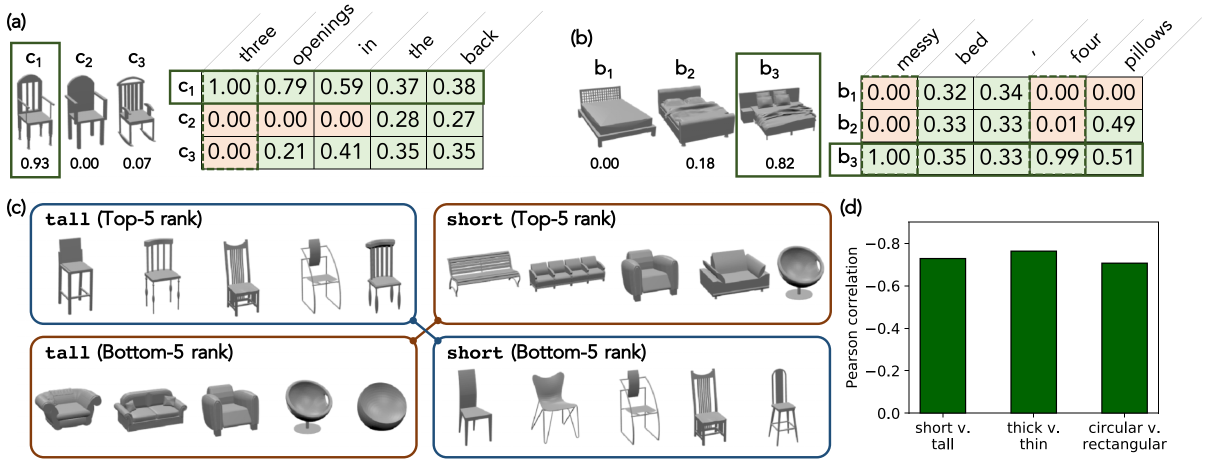
Task B: Zero-Shot Generalization. Our second CiC benchmark task is zero-shot generalization, which examines the ability for the model to generalize from understanding attribute concepts learned in a chairs context to contexts with unseen object classes like tables and lamps. The overall task setting is the same as before, but during evaluation the triplets are composed of objects from a particular unseen class. For consistency with prior work, all models here are evaluated on an image-only setting (i.e., no 3D point-cloud representation). We provide a breakdown of the results on the full zero-shot transfer set by class.

Method	Input	Listener Acc.
Majority	N/A	0.333
*SG-NoAttn	VGG16-SN	0.812 ± 0.008
*SG-Attn	VGG16-SN	0.817 ± 0.008
LSTM-Attn	VGG16-SN	0.731 ± 0.012
PoE	VGG16-SN	0.752 ± 0.009
NMN	VGG16-SN	0.763 ± 0.023
MAC	VGG16-SN	0.818 ± 0.013
NES	VGG16	0.842 ± 0.005
NES	VGG16-SN	**0.856 ± 0.005**
NES	Res101	0.853 ± 0.011
NES^{+}	Res101	**0.870 ± 0.009**

Table 1: **CiC-Language Generalization.** NES on real-world language from the Chairs-in-Context (CiC) dataset. *SG architectures from Achlioptas et al. (2019) are the previously reported state-of-the-art method. NES^{+} grounds sub-events on the feature grid input. -SN indicates ShapeNet pre-trained features.

Models and Implementation. Our main baseline is the recent ShapeGlot (SG) architecture (Achlioptas et al., 2019). The SG baseline leverages recurrent, convolutional, and attention components in an end-to-end architecture to achieve state-of-the-art performance on the language and zero-shot generalization datasets. We also consider a conjunctive baseline with event classifiers without the soft event routing stage, reminiscent of a product-of-experts (PoE) classification setting. This baseline serves to illustrate the impact of the flexible routing stage on compositionality, and in particular handling of non-intersective modifiers. We additionally report two **compositional** baselines from Section 4.1.2, MAC and NMN, following the protocols outlined by our previous end-to-end synthetic experiments 4.1.4. Because CiC contains unstructured human-generated text and it is difficult to train NMN end-to-end from denotation alone, we initialize the sequence-to-sequence program generator in the NMN baseline by pre-training on auxiliary parse information for 1,000 examples (Suhr et al., 2019; Yi et al., 2018); all other baselines do not have any additional supervision data. Finally, we also consider a denser input event space for NES corresponding to sub-regions in the image input. Here, sub-events are additionally sampled from the (unannotated) final *conv4* feature grid of the encoder network; we denote this as NES^{+} in our experiments. We adopt consistent experimental settings from Achlioptas et al. (2019), treating each chair as an event candidate space, with predictions normalized by 3-way softmax over possible target images. All model sizes are kept comparable in number of parameters for fair comparison. We leverage the same pre-trained VGG16 features (Simonyan and Zisserman, 2015; Chang et al., 2015) and GloVe (Wiki.6B) embeddings (Pennington et al., 2014). For completeness, we report results with VGG16 and ResNet-101 *without* ShapeNet pre-training for both tasks.

Analysis. We report our results in Table 1 and Table 2 against the prior state-of-the-art SG architecture (Achlioptas et al., 2019). The MAC baseline provides comparable performance to the prior state-of-the-art. The NMN baseline has reasonable accuracy, albeit lower than the MAC and SG baselines. This is likely due to the ambiguity in longer token sequences (up to 33 tokens), which can contain filler words and occasional disfluencies that hurt the efficacy of the sequence-to-sequence program generator. Nonetheless, NMN outperforms the PoE baseline, which serves



	three	openings	in	the	back
c1	1.00	0.79	0.59	0.37	0.38
c2	0.00	0.00	0.00	0.28	0.27
c3	0.00	0.21	0.41	0.35	0.35

	messy	bed	,	four	pillows
b1	0.00	0.32	0.34	0.00	0.00
b2	0.00	0.33	0.33	0.01	0.49
b3	1.00	0.35	0.33	0.99	0.51

Figure 10: **CiC Qualitative Results.** We visualize results on the **(a)** CiC evaluation set and **(b)** zero-shot evaluation set. Chair and bed triplets (c_*, b_*) are shown with NES output scores. Tables show *relative* classifier scores that are normalized per word, *for visualization purpose only* (e.g., if a word has classifier scores of 1.0 across events, then we show them as 0.333). NES grounds real-world reference language and provides meaningful interpretability on how individual classifiers contribute to the final score. **(c)** Event classifiers can be used standalone for retrieval, showing lexical consistency between antonyms; **(d)** shows Pearson correlations (p-value $< 1\text{e}-13$). See Section 4.2.

	Zero-Shot Classes				
Model	Lamp	Bed	Table	Sofa	All
Major.	0.333	0.333	0.333	0.333	0.333
*SG	0.501	0.564	0.637	0.536	0.560
PoE	0.422	0.466	0.587	0.483	0.490
NMN	0.462	0.492	0.572	0.532	0.515
MAC	0.533	0.531	0.632	0.551	0.567
NES					
w/VGG16	0.544	0.578	0.693	0.588	**0.601**
w/Res101	0.573	0.589	0.715	0.610	**0.622**

Table 2: **CiC-Zero Shot Generalization.** Zero-shot generalization to unseen objects on the Chairs-in-Context (CiC) dataset. Results suggest NES can learn words as event classifiers in a general, object-agnostic manner. *SG model from (Achlioptas et al., 2019).

as a simplistic conjunctive modular baseline without the NES event routing framework.

We observe that our model improves over the prior state-of-the-art work on this dataset by a large margin on the original neural listener task. Further, NES significantly improves zero-shot generalization performance, indicating that it has learned event classifiers for attributes (e.g., ‘‘messy’’, ‘‘tall’’) that can generalize to entirely unseen input event distributions. We visualize qualitative results in Figure 10: NES can provide interpretable event classifier outputs at the word level without any additional low-level supervision, in both the main (chairs) and unseen zero-shot settings. We also show how learned event classifiers are lexically consistent by performing standalone retrieval of antonym pairs. We observe that high-ranked retrievals for a word classifier correlate with low-ranked retrievals of its antonym.

4.3 Overall Discussion

We provide additional discussion of the overall NES framework, considering its broader implications, limitations, and avenues for further work.

Broader Generality. In the above sections, we have described our key results of NES on the ShapeWorld and CiC benchmarks. However, modular neural network approaches like NMN are intuitively suited to settings where the visual and language environments are particularly *regular*, *context-free*, and *unambiguous*. In its current formulation, NES is similarly suited to such structured settings: effective generalization to highly irregular and context-sensitive vision and language settings in images and videos (Zhou et al., 2019), remains outside the current scope of the presented paper. Nonetheless, we believe that careful consideration of some of the key elements in the NES framework, such as the proposed soft event routing system with ungrounded events used for coordinating richer meaning, can offer a promising route towards improving the state-of-the-art.

Computational Complexity. Through its existential quantification operating over events, the complexity of event assignment (Equation (7)) during inference scales by $O(k^{n-1})$, where k is the number of visual event candidates V and $n-1$ the number of events $\mathbf{e}$ in the logical form F (excluding $e_{\emptyset}$). This was not an issue in the domains examined here, but may become one in complex vision-language domains. Exploring potential relationships with concurrent techniques (Bogin et al., 2020) that increase computational complexity but also improve systematicity may prove insightful here as well.

5 Conclusion

In this work, we introduced *neural event semantics* (NES) for compositional grounded language understanding. Our framework's *conjunctivist* design offers a compelling alternative to designs rooted primarily in function-based semantics: By deriving structure from events and their (soft) routings, NES operates with a simpler composition ruleset (conjunction) and effectively learns semantic concepts without any low-level ground truth supervision. Controlled synthetic experiments (ShapeWorld) show the generalization benefits of our framework, and we demonstrate broader applicability of NES on real-world language data (CiC) by significantly improving language and zero-shot generalization over prior state-of-the-art. Ultimately, our work shows that deep consideration of the mechanisms for compositional neural methods may yield techniques better suited for differentiable neural modeling, maintaining core expressivity for grounded language understanding tasks.

Acknowledgments

We sincerely thank Alex Tamkin, Jesse Mu, Mike Wu, Panos Achlioptas, Robert Hawkins, Boris Perkhounkov, Fereshte Khani, Robin Jia, Elisa Kreiss, Jim Fan, Juan Carlos Niebles, Thomas Icard, and Paul Pietroski for helpful discussions and support. We are grateful to our action editor, Luke Zettlemoyer, and our anonymous reviewers for their insightful and constructive feedback during the review process. S. Buch is supported by an NDSEG Fellowship. This work was supported in part by the Office of Naval Research grant ONR MURI N00014-16-1-2007 and by a NSF Expeditions Grant, Award Number (FAIN) 1918771. This article reflects the authors' opinions and conclusions, and not any other entity.

References

Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J. Guibas. 2019. ShapeGlot: Learning language for shape differentiation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8938–8947. https://doi.org/10.1109/ICCV.2019.00903

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. *NAACL*. https://doi.org/10.18653/v1/N16-1181

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48. https://doi.org/10.1109/CVPR.2016.12

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision.* https://doi.org/10.1109/ICCV.2015.279

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019a. Systematic generalization: What is required and can it be learned? *ICLR*.

Dzmitry Bahdanau, Harm de Vries, Timothy J. O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. 2019b. CLOSURE: Assessing systematic generalization of CLEVR models. *arXiv preprint arXiv:1912.05783*.

Ben Bogin, Sanjay Subramanian, Matt Gardner, and Jonathan Berant. 2020. Latent compositional representations improve systematic generalization in grounded question answering. *arXiv preprint arXiv:2007.00266*. https://doi.org/10.1162/tacl_a_00361

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2018. Gated-attention architectures for task-oriented language grounding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv: 1504.00325*.

Ann Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszyńska. 2016. Resources for building applications with dependency minimal recursion semantics. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1240–1247.

Donald Davidson. 1967. The logical form of action sentences. In *The Logic of Decision and Action*, pages 81–120. Pittsburgh: University of Pittsburgh Press.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6): 602–610. `https://doi.org/10.1016/j.neunet.2005.06.042`, Pubmed: 16112549

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2018. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709. `https://doi.org/10.1109/CVPR.2019.00686`

T. S. Jayram, Vincent Marois, Tomasz Kornuta, Vincent Albouy, Emre Sevgen, and Ahmet S. Ozcan. 2019. Transfer learning in visual and relational reasoning. *arXiv preprint arXiv:1911.11938*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017a. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1997. IEEE. `https://doi.org/10.1109/CVPR.2017.215`

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017b. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3008–3017. `https://doi.org/10.1109/ICCV.2017.325`

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 3128–3137. https://doi.org/10.1109/CVPR.2015.7298932

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206. https://doi.org/10.1162/tacl_a_00220

Jayant Krishnamurthy, Oyvind Tafjord, and Aniruddha Kembhavi. 2016. Semantic parsing to probabilistic programs for situated question answering. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 160–170, Austin, Texas. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1016

Alexander Kuhnle and Ann A. Copestake. 2017. ShapeWorld - A new test methodology for multimodal language understanding. *CoRR*, abs/1704.04517.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40. https://doi.org/10.1017/S0140525X16001837, Pubmed: 27881212

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. *Advances in neural information processing systems*, 27:1682–1690.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

Vincent Marois, T. S. Jayram, Vincent Albouy, Tomasz Kornuta, Younes Bouhadjar, and Ahmet S. Ozcan. 2018. On transfer learning using a MAC model variant. *arXiv preprint arXiv:1811.06529*.

Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. *International Conference on Machine Learning*.

Will Monroe, Robert XD Hawkins, Noah D. Goodman, and Christopher Potts. 2017. Colors in Context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338. https://doi.org/10.1162/tacl_a_00064

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398. https://doi.org/10.1111/j.1755-2567.1970.tb00434.x

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Paul M. Pietroski. 2005. *Events and Semantic Architecture*. Oxford University Press.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. 2020. A benchmark for systematic generalization in grounded language understanding. *Advances in Neural Information Processing Systems*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5594–5608. https://doi.org/10.18653/v1/2020.acl-main.495

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A Corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428. https://doi.org/10.18653/v1/P19-1644

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3–4):229–256. https://doi.org/10.1007/BF00992696

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. 2019. Grounded video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6578–6587. https://doi.org/10.1109/CVPR.2019.00674

A Appendix: Equation 8

We describe a formula for an approximation of the max function (L_∞-norm), used during training to improve end-to-end gradient flow (Section 3.6). Let $\mathrm{s} \in \mathbb{R}^n \geq 0$ be a vector of *non-negative* scores. We consider $f_{max} : \mathbb{R}^n \to \mathbb{R}$ (Equation (8) in Section 3.6):

$$f_{max}(\mathbf{s}; \beta) = \frac{\sum_i (\mathbf{s}_i)^{\beta+1}}{\sum_i (\mathbf{s}_i)^{\beta}}$$

where $\beta \geq 0$ is a hyperparameter. As $\beta \to \infty$, $f_{max}(\mathbf{s}; \beta) \to s^*$, where $s^* = \max(\mathbf{s}) = L_\infty(\mathbf{s})$. We can show this by dividing the numerator and denominator by $(s^*)^{\beta+1}$ and taking the limit:

$$\lim_{\beta\to\infty} f_{max}(\mathbf{s}) = \lim_{\beta\to\infty} \left(\frac{\sum_i (\mathbf{s}_i)^{\beta+1} / (s^*)^{\beta+1}}{\sum_i (\mathbf{s}_i)^{\beta} / (s^*)^{\beta+1}} \right) \tag{9}$$

Now all terms where $|\mathbf{s}_i| < s^*$ tend to 0, leaving us just the maximum terms in the numerator and denominator where $|\mathbf{s}_i| = s^*$. Thus, Equation (9) reduces to $\frac{1}{1/s^*} = \boxed{s^*}$, as desired.

Further, $|f_{max}(\mathbf{s})|$ has the essential property of *always being upper-bounded* by s^*. We show this by Hölder's inequality. Let $x_i = \mathbf{s}_i, y_i = (\mathbf{s}_i)^\beta$, and let $p \to \infty$ and $q \to 1$ (satisfying conditions $1/p + 1/q = 1$ and $p, q \in (1, \infty)$). Then,

$$\begin{aligned} \sum_i |x_i y_i| &\leq \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} \left(\sum_i |y_i|^q\right)^{\frac{1}{q}} \\ \frac{\sum_i |x_i y_i|}{\left(\sum_i |y_i|^q\right)^{\frac{1}{q}}} &\leq \left(\sum_i |x_i|^p\right)^{\frac{1}{p}} \\ |f_{max}(\mathbf{s})| &\leq L_\infty(\mathbf{s}) = \boxed{s^*} \end{aligned}$$

Thus, with non-negative scores $\mathbf{s}_i \geq 0$, we have $\lim_{\beta\to\infty} f_{max}(\mathbf{s}) = \max(\mathbf{s})$ and $f_{max}(\mathbf{s}) \leq \max(\mathbf{s})$.