Grounding Spatio-Temporal Language with Transformers

Tristan Karch[*], **Laetitia Teodorescu**[*]
Inria - Flowers Team
Université de Bordeaux
firstname.lastname@inria.fr

Katja Hofmann
Microsoft Research
Cambridge, UK

Clément Moulin-Frier
Inria - Flowers team
Université de Bordeaux
ENSTA ParisTech

Pierre-Yves Oudeyer
Inria - Flowers team
Université de Bordeaux
ENSTA ParisTech

Abstract

Language is an interface to the outside world. In order for embodied agents to use it, language must be grounded in other, sensorimotor modalities. While there is an extended literature studying how machines can learn grounded language, the topic of how to learn spatio-temporal linguistic concepts is still largely uncharted. To make progress in this direction, we here introduce a novel spatio-temporal language grounding task where the goal is to learn the meaning of spatio-temporal descriptions of behavioral traces of an embodied agent. This is achieved by training a truth function that predicts if a description matches a given history of observations. The descriptions involve time-extended predicates in past and present tense as well as spatio-temporal references to objects in the scene. To study the role of architectural biases in this task, we train several models including multimodal Transformer architectures; the latter implement different attention computations between words and objects across space and time. We test models on two classes of generalization: 1) generalization to randomly held-out sentences; 2) generalization to grammar primitives. We observe that maintaining object identity in the attention computation of our Transformers is instrumental to achieving good performance on generalization overall, and that summarizing object traces in a single token has little influence on performance. We then discuss how this opens new perspectives for language-guided autonomous embodied agents. We also release our code under open-source license as well as pretrained models and datasets to encourage the wider community to build upon and extend our work in the future.

1 Introduction

Building autonomous agents that learn to represent and use language is a long standing-goal in *Artificial Intelligence* [19, 38]. In developmental robotics [6], language is considered a cornerstone of development that enables embodied cognitive agents to learn more efficiently through social interactions with humans or through cooperation with other autonomous agents. It is also considered essential to develop complex sensorimotor skills, facilitating the representation of behaviors and actions.

Embodied Language Grounding [47] is the field that studies how agents can align language with their behaviors in order to extract the meaning of linguistic constructions. Early approaches in

[*] Equal Contribution

Preprint. Under review.

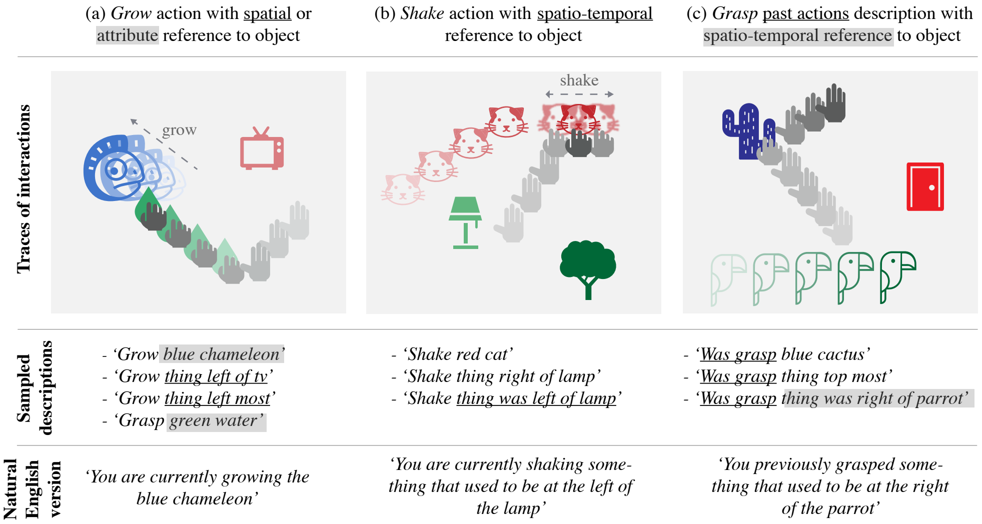

(a) Grow action with spatial or attribute reference to object
(b) Shake action with spatio-temporal reference to object
(c) Grasp past actions description with spatio-temporal reference to object
Traces of interactions
grow
shake
Sampled descriptions
- 'Grow blue chameleon'
- 'Grow thing left of tv'
- 'Grow thing left most'
- 'Grasp green water'
- 'Shake red cat'
- 'Shake thing right of lamp'
- 'Shake thing was left of lamp'
- 'Was grasp blue cactus'
- 'Was grasp thing top most'
- 'Was grasp thing was right of parrot'
Natural English version
'You are currently growing the blue chameleon'
'You are currently shaking something that used to be at the left of the lamp'
'You previously grasped something that used to be at the right of the parrot'

Figure 1: **Visual summary of the Temporal Playground environment:** At each episode (column a, b and c), the actions of an agent (represented by a hand) unfold in the environment and generate a trace of interactions between objects and the agent body. Given such a trace, the environment automatically generates a set of synthetic linguistic descriptions that are true at the end of the trace. In (a) the agent grows an object which is described with spatial (underlined) or attribute (highlighted) reference. In (b) it shakes an object which is described with attribute, spatial or spatio-temporal (underlined) reference. In (c) it has grasped an object (past action underlined) which is described with attribute, spatial or spatio-temporal (highlighted) reference.

developmental robotics studied how various machine learning techniques, ranging from neural networks [39, 42, 23] to non-negative matrix factorization [31], could enable the acquisition of grounded compositional language [41, 40]. This line of work was recently extended using techniques for *Language conditioned Deep Reinforcement Learning* [30]. Among these works we can distinguish mainly three language grounding strategies. The first one consists of directly grounding language in the behavior of agents by training goal-conditioned policies satisfying linguistic instructions [39, 42, 22, 21, 8]. The second aims at extracting the meaning of sentences from mental simulations (i.e. generative models) of possible sensorimotor configurations matching linguistic descriptions [31, 1, 11, 33]. The third strategy searches to learn the meaning of linguistic constructs in terms of outcomes that agents can observe in the environment. This is achieved by training a truth function that detects if descriptions provided by an expert match certain world configurations. This truth function can be obtained via *Inverse Reinforcement Learning* [46, 2] or by training a multi-modal binary classifier [13].

While all the above-mentioned approaches consider language that describes immediate and instantaneous actions, we argue that it is also important for agents to grasp language expressing concepts that span multiple time scales. We thus propose to study the grounding of new spatio-temporal concepts enabling agents to ground time extended predicates (Fig. 1a) with complex spatio-temporal references to objects (Fig. 1b) and understand both present and past tenses (Fig. 1c). To do so we choose the third strategy mentioned above, i.e. to train a truth function that predicts when descriptions match traces of experience. This choice is motivated by two important considerations. First, prior work showed that learning truth functions was key to foster generalization [2], enabling agents to efficiently self-train policies via goal imagination [13] and goal relabeling [11]. Hence the truth function is an important and self-contained component of larger learning systems. Second, this strategy allows to carefully control the distribution of experiences and descriptions perceived by the agent.

Grounding spatio-temporal language is a relational problem. In the context of this paper, the concepts we aim at grounding are temporal and spatial, and thus relational by nature. But more generally, it is worth mentioning that Embodied Language Grounding has a relational structure. We understand the meaning of words by analyzing the relations they state in the world [17]. Actions are relations between subjects and objects, and can be defined in terms of affordances of the agent [18]. As a result we implement our truth function using relational architectures based on *Transformers* [43] and investigate the role of the relational bias [3] on learning. We propose a formalism unifying three variants of a multi-modal transformer inspired by Ding et al. [15] that implement different relational operations. We measure the generalization capabilities of these architectures along three axis 1) generalization to new traces of experience; 2) generalization to randomly held out sentences; 3) generalization to grammar primitives, systematically held out from the training set as in Ruis et al. [36]. We observe that maintaining object identity in the attention computation of our Transformers is instrumental to achieving good performance on generalization overall. We also identify specific relational operation that are key to generalize on certain grammar primitives.

Contributions. This paper introduces:

1. A new Embodied Language Grounding task focusing on spatio-temporal language;
2. A formalism unifying different relational architectures based on Transformers expressed as a function of mapping and aggregation operations;
3. A systematic study of the generalization capabilities of these architectures and the identification of key components for their success on this task.

2 Methods

2.1 Problem Definition

We consider the setting of an embodied agent behaving in an environment. This agent interacts with the surrounding objects over time, during an episode of fixed length (T). Once this episode is over, an oracle provides exhaustive feedback in a synthetic language about everything that has happened. This language describes actions of the agent over the objects and includes spatial and temporal concepts. The spatial concepts are reference to an object through its spatial relation with others (Fig. 1a), and the temporal concepts are the past modality for the actions of the agent (Fig. 1c), past modality for spatial relations (Fig. 1b), and actions that unfold over time intervals. The histories of states of the agent's body and of the objects over the episode as well as the associated sentences are recorded in a buffer $\mathcal{B}$. From this setting, and echoing previous work on training agents from descriptions, we frame the Embodied Language Grounding problem as learning a parametrized truth function R_θ over couples of observations traces and sentences, tasked with predicting whether a given sentence W is true of a given episode history S or not. Formally, we aim to minimize

$$\mathbb{E}_{(S,W)\sim\mathcal{B}}\big[\mathcal{L}(R_\theta(S,W), r(S,W))\big]$$

where $\mathcal{L}$ denotes the cross-entropy loss and r denotes the ground truth boolean value for sentence W about trace S.

2.2 Temporal Playground

In the absence of any dedicated dataset providing spatio-temporal descriptions from behavioral traces of an agent, we introduce *Temporal Playground* (Fig. 1) an environment coupled with a templated grammar designed to study spatio-temporal language grounding. The environment is a 2D world, with procedurally-generated scene containing $N = 3$ objects sampled from 32 different object types belonging to 5 categories. Each object has a continuous 2D position, a size, a continuous color code specified by a 3D vector in RGB space, a type specified by a one-hot vector, and a boolean unit specifying whether it is grasped. The size of the object feature vector (o) is $|o| = 39$. The agent's body has its 2D position in the environment and its gripper state (grasping or non-grasping) as features (body feature vector (b) of size $|b| = 3$). In this environment, the agent can perform various actions over the length (T) of an episode. Some of the objects (the animals) can move independently. Objects can also interact: if the agent brings food or water to an animal, it will grow in size; similarly, if water is brought to a plant, it will grow. At the end of an episode, a generative grammar generates sentences describing all the interactions that occurred. A complete specification of the environment as well as the BNF of the grammar can be found in Supplementary Section 6.2.

Synthetic language. To enable a controlled and systematic study of how different types of spatio-temporal linguistic meanings can be learned, we argue it is necessary to first conduct a systematic study with a controlled synthetic grammar. We thus consider a synthetic language with a vocabulary of size 53 and sentences with a maximum length of 8. This synthetic language facilitates the generation of descriptions matching behavioral traces of the agent. Moreover, it allows us to express four categories of concepts associated with specific words. Thus, the generated sentences consist in four conceptual types based on the words they involve:

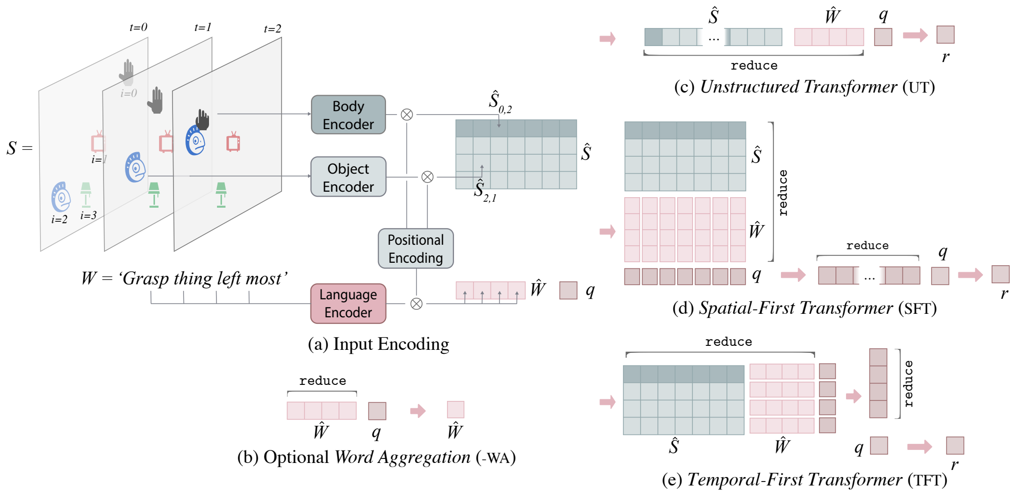
- **Sentences involving basic concepts.** This category of sentences talk about present-time events by referring to objects and their attributes. Sentences begin with the *'grasp'* token combined with any object. Objects can be named after their category (eg. *'animal', 'thing'*) or directly by their type (*'dog', 'door', 'algae', etc.*). Finally, the color *('red','blue','green')* of objects can also be specified.
- **Sentences involving spatial concepts.** This category of sentences additionally involve one-to-one spatial relations and one-to-all spatial relations to refer to objects. An object can be *'left of'* another object (reference is made in relation to a single other object), or can be the *'top most'* object (reference is made in relation with all other objects). Example sentences include *'grasp thing bottom of cat'* or *'grasp thing right most'*.
- **Sentences involving temporal concepts**. This category of sentences involves talking about temporally-extended predicates and the past tense, without any spatial relations. The two temporal predicates are denoted with the words *'grow'* and *'shake'*. The truth value of these predicates can only be decided by looking at the temporal evolution of the object's size and position respectively. A predicate is transposed at the past tense if the action it describes was true at some point in the past and is no longer true in the present, this is indicated by adding the modifier *'was'* before the predicate. Example sentences include *'was grasp red chameleon'* (indicating that the agent grasped the red chameleon and then released it) and *'shake bush'*;
- **Sentences involving spatio-temporal concepts**. Finally, we consider the broad class of spatio-temporal sentences that combine spatial reference and temporal or past-tense predicates. These are sentences that involve both the spatial and temporal concepts defined above. Additionally, there is a case of where the spatial and the temporal aspects are entangled: past spatial reference. This happens when an object is referred to by its previous spatial relationship with another object. Consider the case of an animal that was at first on the bottom of a table, then moved on top, and then is grasped. In this case we could refer to this animal as something that was previously on the bottom of the table. We use the same *'was'* modifier as for the past tense predicates; and thus we would describe the action as *'Grasp thing was bottom of table'*.

2.3 Architectures

In this section we describe the architectures used as well as their inputs. Let one input sample to our model be $I = (S, W)$, where $(S_{i,t})_{i,t}$ represents the objects' and body's evolution, and $(W_l)_l$ represents the linguistic observations. S has a spatial (or entity) dimension indexed by $i \in [0..N]$ and a temporal dimension indexed by $t \in [1..T]$; for any i, t, $S_{i,t}$ is a vector of observational features. Note that by convention, the trace $(S_{0,t})_t$ represents the body's features, and the traces $(S_{i,t})_{t,i>0}$ represents the other objects' features. W is a 2-dimensional tensor indexed by the sequence $l \in [1..L]$; for any l, $W_l \in \mathbb{R}^{d_W}$ is a one-hot vector defining the word in the dictionary. The output to our models is a single scalar between 0 and 1 representing the probability that the sentence encoded by W is true in the observation trace S.

Transformer Architectures To systematically study the influence of architectural choices on language performance and generalization in our spatio-temporal grounded language context, we define a set of mapping and aggregation operations that allows us to succinctly describe different models in a unified framework. We define:

- An aggregation operation based on a Transformer model, called `reduce`. `reduce` is a parametrized function that takes 3 inputs: a tensor, a dimension tuple D over which to reduce and a query tensor (that has to have the size of the reduced tensor). R layers of a Transformer are applied to the input-query concatenation and are then queried at the position corresponding to the query tokens. This produces an output reduced over the dimensions D.



(a) Input Encoding

(b) Optional *Word Aggregation* (-WA)

(c) *Unstructured Transformer* (UT)

(d) *Spatial-First Transformer* (SFT)

(e) *Temporal-First Transformer* (TFT)

Figure 2: **Visual summary of the architectures used.** We show the details of UT, SFT and TFT respectively in subfigures (c), (d), (e), as well as a schematic illustration of the preprocessing phase (a) and the optional word-aggregation procedure (b).

- A casting operation called `cast`. `cast` takes as input 2 tensors A and B and a dimension d. A is flattened, expanded so as to fit the tensor B in all dimensions except d, and concatenated along the d dimension.
- A helper expand operation called `expand` that takes as arguments a tensor and an integer n and repeats the tensor n times.

Using those operations, we define three architectures: one with no particular bias (*Unstructured Transformer*, inspired by Ding et al. [15], or UT); one with a spatial-first structural bias – objects and words are aggregated along the spatial dimension first (*Spatial-First Transformer* or SFT); and one with a temporal-first structural bias – objects and words are aggregated along the temporal dimension first (*Temporal-First Transformer*, or TFT).

Before inputting the observations of bodies and objects S and the language W into any of the Transformer architectures, they are projected to a common dimension (see Supplementary Section 7.2 for more details). A positional encoding [43] is then added along the time dimension for observations and along the sequence dimension for language; and finally a one-hot vector indicating whether the vector is observational or linguistic is appended at the end. This produces the modified observation-language tuple $(\hat{S}, \hat{W})$. We let:

$$\text{UT}(\hat{S}, \hat{W}) := \texttt{reduce}(\texttt{cast}(\hat{S}, \hat{W}, 0), 0, q)$$

$$\text{SFT}(\hat{S}, \hat{W}, q) := \texttt{reduce}(\texttt{reduce}(\texttt{cast}(\hat{W}, \hat{S}, 0), 0, \texttt{expand}(q, T)), 0, q)$$

$$\text{TFT}(\hat{S}, \hat{W}, q) := \texttt{reduce}(\texttt{reduce}(\texttt{cast}(\hat{W}, \hat{S}, 1), 1, \texttt{expand}(q, N + 1)), 0, q)$$

where T is the number of time steps, N is the number of objects and q is a learned query token. See Fig. 2 for an illustration of these architectures.

Note that SFT and TFT are transpose versions of each other: SFT is performing aggregation over space first and then time, and the reverse is true for TFT. Additionally, we define a variant of each of these architectures where the words are aggregated before being related with the observations. We name these variants by appendding -WA (word-aggregation) to the name of the model (see Fig. 2 (b)).

$$\hat{W} \leftarrow \texttt{reduce}(\hat{W}, 0, q)$$

We examine these variants to study the effect of letting word-tokens directly interact with object-token through the self-attention layers vs simply aggregating all language tokens in a single embedding and letting this vector condition the processing of observations. The latter is commonly done in the language-conditioned RL and language grounding literature [10, 2, 24, 36], using the language embedding in FiLM layers [35] for instance. Finding a significant effect here would encourage using architectures which allow direct interactions between the word tokens and the objects they refer to.

LSTM Baselines We also compare some LSTM-based baselines on this task; their architecture is described in more detail in Supplementary Section 7.3.

2.4 Data Generation, Training and Testing Procedures

We use a bot to generate the episodes we train on. The data collected consists of 56837 trajectories of $T = 30$ time steps. Among the traces some descriptions are less frequent than others but we make sure to have at least 50 traces representing each of the 2672 descriptions we consider. We record the observed episodes and sentences in a buffer, and when training a model we sample (S, W, r) tuples with one observation coupled with either a true sentence from the buffer or another false sentence generated from the grammar. More details about the data generation can be found in Supplementary Section 7.1.

For each of the Transformer variants (6 models) and the LSTM baselines (2 models) we perform an hyper parameter search using 3 seeds in order to extract the best configuration. We extract the best condition for each model by measuring the mean F_1 on a testing set made of uniformly sampled descriptions from each of the categories define in section 2.2. We use the F_1 score because testing sets are imbalanced (the number of traces fulfilling each description is low). We then retrain best configurations over 10 seeds and report the mean and standard deviation (reported as solid black lines in Fig. 3 and Fig. 4) of the averaged F_1 score computed on each set of sentences. When statistical significance is reported in the text, it is systematically computed using a two-tail Welch's t-test with null hypothesis $\mu_1 = \mu_2$, at level $\alpha = 0.05$ [12]. Details about the training procedure and the hyper parameter search are provided in Supplementary Section 7.4.

3 Experiments and Results

3.1 Generalization abilities of models on non-systematic split by categories of meaning

In this experiment, we perform a study of generalization to new sentences from known observations. We divide our set of test sentences in four categories based on the categories of meanings listed in Section 2.2: Basic, Spatial, Spatio-Temporal and Temporal. We remove 15% of all possible sentences in each category from the train set and evaluate the F1 score on those sentences. The results are provided in Fig. 3.

First, we notice that over all categories of meanings, all UT and TFT models, with or without word-aggregation, perform extremely well compared to the LSTM baselines, with all these four models achieving near-perfect test performance on the Basic sentences, with very little variability across the 10 seeds. We then notice that all SFT variants perform poorly on all test categories, in line or worse than the baselines. This is particularly visible on the spatio-temporal category, where the SFT models perform at 0.75 ± 0.020 whereas the baselines perform at 0.80 ± 0.019 . This suggests that across tasks, it is harmful to aggregate each scene plus the language information into a single vector. This may be due to the fact that objects lose their identity in this process, since information about all the objects becomes encoded in the same vector. This may make it difficult for the network to perform computations about the truth value of predicate on a single object.

Secondly, we notice that the word-aggregation condition seems to have little effect on the performance on all three Transformer models. We only observe a significant effect for UT models on spatio-temporal concepts (p-value $=$ 2.38e-10). This suggests that the meaning of sentences can be adequately summarised by a single vector; while maintaining separated representations for each object is important for achieving good performance it seems unnecessary to do the same for linguistic input. However we notice during our hyperparameter search that our -WA models are not very robust to hyperparameter choice, with bigger variants more sensitive to the learning rate.

Thirdly, we observe that for our best-performing models, the basic categories of meanings are the easiest, with a mean score of 1.0 ± 0.003 across all UT and TFT models, then the spatial ones at 0.96 ± 0.020, then the temporal ones at 0.96 ± 0.009, and finally the spatio-temporal ones at 0.89 ± 0.027. This effectively suggests, as we hypothesised, that sentences containing spatial relations or temporal concepts are harder to ground than those who do not.

Known sentences with novel observations We also examine the mean performance of our models for sentences in the training set but evaluated on a set of *new observations*: we generate a new set of rollouts on the environment, and only evaluate the model on sentences seen at train time (plots are reported in Supplementary Section 8). We see the performance is slightly better in this case, especially for the LSTM baselines (0.82 ± 0.031 versus 0.79 ± 0.032), but the results are comparable in both cases, suggesting that the main difficulty for models lies in grounding spatio-temporal meanings and not in linguistic generalization for the type of generalization considered in this section.

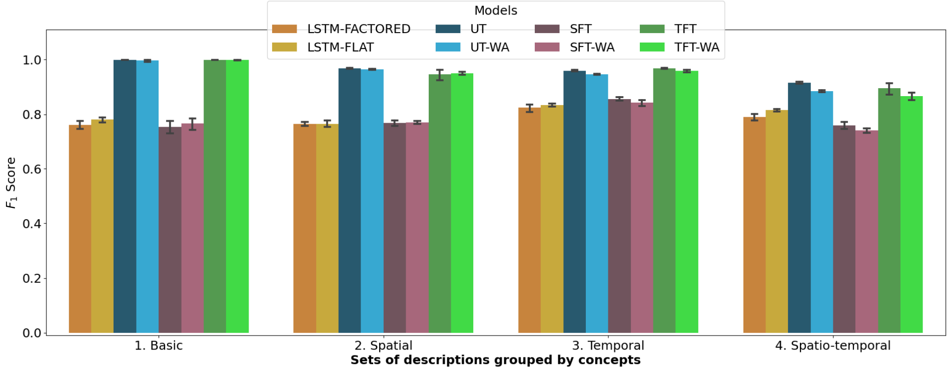


Figure 3: **F1 scores for all the models on randomly held-out sentences.** F_1 is measured on separated sets representing each category of concepts defined in Section 2.2.

3.2 Systematic generalization on withheld combinations of words

In addition to the previous generalization studies, we perform an experiment in a harder linguistic generalization setting where we systematically remove binary combinations in our train set. This is in line with previous work on systematic generalization on deep learning models [28, 36, 25]. We create five test sets to examine the abilities of our models to generalize on binary combinations of words that have been systematically removed from the set of training sentences, but whose components have been seen before in other contexts. Our splits can be described by the set of forbidden combinations of words as:

1. **Forbidden object-attribute combinations.** remove from the train set all sentences containing *'red cat'*, *'blue door'* and *'green cactus'*. This tests the ability of models to recombine known objects with known attributes;
2. **Forbidden predicate-object combination.** remove all sentences containing *'grow'* and all objects from the *'plant'* category. This tests the model's ability to apply a known predicate to a known object in a new combination;
3. **Forbidden one-to-one relation.** remove all sentences containing *'right of'*. Since the *'right'* token is already seen as-is in the context of one-to-all relations (*'right most'*), and other one-to-one relations are observed during training, this tests the abilities of models to recombine known directions with in a known template;
4. **Forbidden past spatial relation.** remove all sentences containing the contiguous tokens *'was left of'*. This tests the abilities of models to transfer a known relation to the past modality, knowing other spatial relations in the past;
5. **Forbidden past predicate.** remove all sentences containing the contiguous tokens *'was grasp'*. This tests the ability of the model to transfer a known predicate to the past modality, knowing that it has already been trained on other past-tense predicates.

To avoid retraining all models for each split, we create one single train set with all forbidden sentences removed and we test separately on all splits. We use the same hyperparameters for all models than in the previous experiments. The results are reported in Fig. 4.

First we can notice that the good test scores obtained by the UT and TFT models on the previous sections are confirmed in on this experiment: they are the best performing models overall. We then notice that the first two splits, corresponding to new attribute-object and predicate-object combinations, are solved by the UT and TFT models, while the SFT models and the LSTM baselines struggle to achieve high scores. For the next 3 splits, which imply new spatial and temporal combinations, the scores overall drop significantly; we also observe much wider variability between seeds for each model, perhaps suggesting the various strategies adopted by the models to fit the train set have very different implications in terms of systematic generalization on spatial and temporal concepts. This very high variability between seeds on systematic generalization scores are reminiscent of the results obtained on the gSCAN benchmark [36].

Additionally, for split 3, which implies combining known tokens to form a new spatial relation, we observe a significant drop in generalization for the word-aggregation (WA) conditions, consistent across models (on average across seeds, -0.14 ± 0.093, -0.15 ± 0.234 and -0.20 ± 0.061 for UT, SFT and TFT resp. with p-values $<$ 1e-04 for UT and SFT). This may be due to the fact that recombining any one-to-one relation with the known token *right* seen in the context of one-to-all relations requires a separate representation for each of the linguistic tokens. The same significant drop in performance for the WA condition can be observed for UT and TFT in split 4, which implies transferring a known spatial relation to the past.

However, very surprisingly, for split 5 – which implies transposing the known predicate *grasp* to the past tense – we observe a very strong effect in the opposite direction: the WA condition seems to help generalizing to this unknown past predicate (from close-to-zero scores for all transformer models, the *WA* adds on average 0.71 ± 0.186, 0.45 ± 0.178 and 0.52 ± 0.183 points for UT, ST and TT resp. and p-values$<$ 1e-05). This may be due to the fact that models without WA learn a direct and systematic relationship between the *grasp* token and grasped objects, as indicated in their features; this relation is not modulated by the addition of the *was* modifier as a prefix to the sentence. Models do not exhibit the same behavior on split 4, which has similar structure (transfer the relation *left of* to the past). This may be due to the lack of variability in instantaneous predicates (only the *grasp* predicate); whereas there are several spatial relations (4 one-to-one, 4 one-to-all).

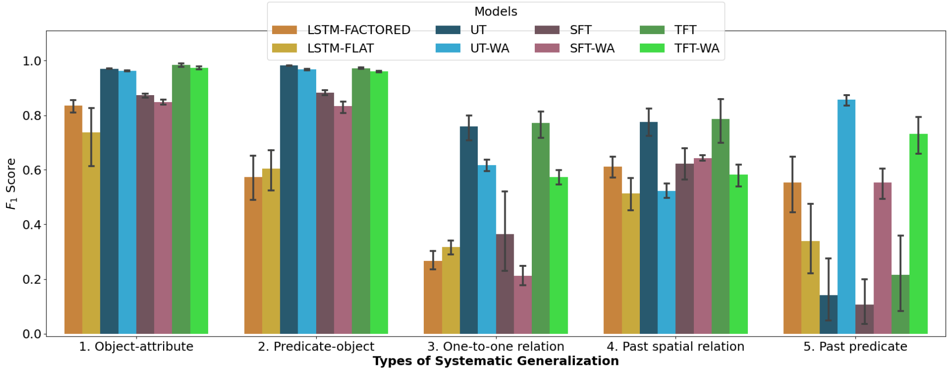


Figure 4: **$\mathbf{F_1}$ scores of all the models on systematic generalization splits.** F_1 is measured on separated sets representing each of the forbidden combinations of word defined above.

4 Related Work

The idea that agents should learn to represent and ground language in their experience of the world has a long history in developmental robotics [47, 38, 39, 6] and was recently extended in the context of Language Conditioned Deep Reinforcement Learning [10, 21, 30, 2]. These recent approaches often consider navigation [9, 8] or object manipulation [1, 21] tasks and are always using instructive language. Meanings typically refer to instantaneous actions and rarely consider spatial reference to objects [34]. Although our environment includes object manipulations, we here tackle novel categories of meanings involving the grounding of spatio-temporal concepts such as the past modality or complex spatio-temporal reference to objects.

We evaluate our learning architectures on their ability to generalise to sets of descriptions that contain systematic differences with the training data so as to assess whether they correctly model grammar primitives. This procedure is similar to the *gSCAN* benchmark [36]. This kind of compositional generalisation is referred as 'systematicity' by Hupkes et al. [25]. Environmental drivers that facilitate systematic generalization are also studied by Hill et al. [22]. Although Hupkes et al. [25] consider relational models in their work, they do not evaluate their performance on a *Language Grounding* task. Ruis et al. [36] consider an Embodied Language Grounding setup involving one form of time-extended meanings (adverbs), but do not consider the past modality and spatio-temporal reference to objects, and do not consider learning truth functions. Also, they do not consider learning architectures that process sequences of sensorimotor observations. To our knowledge, no previous work has conducted systematic generalization studies on an Embodied Language Grounding task involving spatio-temporal language with Transformers.

The idea that relational architectures are relevant models for Language Grounding has been previously explored in the context of *Visual Reasoning*. They were indeed successfully applied for spatial reasoning in the visual question answering task *CLEVR* [37]. With the recent publication of the video reasoning dataset *CLEVRER* [44], those models were extended and demonstrated abilities to reason over spatio-temporal concepts, correctly answering causal, predictive and counterfactual questions [15]. In contrast to our study, these works around CLEVRER do not aim to analyze spatio-temporal language and therefore do not consider time-extended predicates or spatio-temporal reference to objects in their language, and do not study properties of systematic generalization over sets of new sentences.

5 Discussion and Conclusion

In this work, we have presented a first step towards learning Embodied Language Grounding of spatio-temporal concepts, framed as the problem of learning a truth function that can predict if a given sentence is true of temporally-extended observations of an agent interacting with a collection of objects. We have studied the impact of architectural choices on successful grounding of our artificial spatio-temporal language. We have modelled different possible choices for aggregation of observations and language as hierarchical Transformer architectures. We have demonstrated that in our setting, it is beneficial to process temporally-extended observations and language tokens side-by-side, as evidenced by the good score of our Unstructured Transformer variant. However, there seems to be only minimal effect on performance in aggregating temporal observations along the temporal dimension first – compared to processing all traces and the language in an unstructured manner – as long as object identity is preserved. This can inform architectural design in cases where longer episode lengths make it impossible to store all individual timesteps for each object; our experiments provide evidence that a temporal summary can be used in these cases. Our experiments with systematic dimensions of generalization provide mixed evidence for the influence of summarizing individual words into a single vector, showing it can be detrimental to generalize to novel word combinations but also can help prevent overgeneralization of a relation between a single word and a single object without considering the surrounding linguistic context.

Limitations and further work. There are several limitations of our setup which open important opportunities for further work. First, we have used a synthetic language that could be extended: for instance with more spatial relations and relations that are more than binary. Another axis for further research is using low-level observations. In our setting, we wanted to disentangle the effect of structural biases on learning spatio-temporal language from the problem of extracting objects from low level observations [5, 20, 16, 29, 7] in a consistent manner over time (object permanence [14, 45]). Further steps in this direction are needed, and it could allow us to define richer attributes (related to material or texture) and richer temporal predicates (such as breaking, floating, etc). Finally, we use a synthetic language which is far from the richness of the natural language used by humans, but previous work has shown that natural language can be projected onto the subspace defined by synthetic language using the semantic embeddings learned by large language models [32]: this opens up be a fruitful avenue for further investigation. A further interesting future work would be to equip autotelic agents, i.e. agents that learn to represent and pursue their on goals using compositional language [13] with our proposed multi-modal transformer architecture so as to autonomously explore their environments by imagining goal narratives [4] made of complex spatio-temporal meanings. More precisely, the agent could collect narratives by interacting with a Social Partner and align it with its recent history of observations. The agent could then use this aligned data to learn a truth function that could serve to self-train on policies matching new imagined narratives.

Broader Impact This work provides a step in the direction of building agents that better understand how language relates to the physical world; this can lead to personal robots that can better suit the needs of their owners because they can be interacted with using language. If successfully implemented, this technology can raise issues concerning automation of certain tasks resulting in loss of jobs for less-qualified workers.

References

[1] Ahmed Akakzia, Cédric Colas, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Grounding Language to Autonomously-Acquired Skills via Goal Generation. In *ICLR 2021 - Ninth International Conference on Learning Representation*, Vienna / Virtual, Austria, May 2021. URL `https://hal.inria.fr/hal-03121146`.

[2] Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward, 2019.

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

[4] Jerome Bruner. The narrative construction of reality. *Critical inquiry*, 18(1):1–21, 1991.

[5] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019.

[6] Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Francesco Nori, Luciano Fadiga, Britta Wrede, Katharina Rohlfing, Elio Tuci, Kerstin Dautenhahn, Joe Saunders, and Arne Zeschel. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3):167–195, 2010. doi: 10.1109/TAMD.2010.2053034.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[8] Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding, 2018.

[9] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 859–865. AAAI Press, 2011.

[10] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning, 2019.

[11] Geoffrey Cideron, Mathieu Seurin, Florian Strub, and Olivier Pietquin. Higher: Improving instruction following with hindsight generation for experience replay. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 225–232, 2020. doi: 10.1109/SSCI47803.2020.9308603.

[12] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms, 2019.

[13] Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Ford Dominey, and Pierre-Yves Oudeyer. Language as a cognitive tool to imagine goals in curiosity-driven exploration, 2020.

[14] Antonia Creswell, Kyriacos Nikiforou, Oriol Vinyals, Andre Saraiva, Rishabh Kabra, Loic Matthey, Chris Burgess, Malcolm Reynolds, Richard Tanburn, Marta Garnelo, and Murray Shanahan. Alignnet: Unsupervised entity alignment, 2020.

[15] David Ding, Felix Hill, Adam Santoro, and Matt Botvinick. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures, 2020.

[16] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations, 2020.

[17] Dedre Gentner and Jeffrey Loewenstein. Relational language and relational thought, 2002.

[18] James J. (James Jerome) Gibson. *The senses considered as perceptual systems*. Allen & Unwin, London, 1968.

[19] Arthur M. Glenberg and Michael P. Kaschak. Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565, September 2002. ISSN 1531-5320. doi: 10.3758/BF03196313. URL `https://doi.org/10.3758/BF03196313`.

[20] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference, 2020.

[21] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, Marcus Wainwright, Chris Apps, Demis Hassabis, and Phil Blunsom. Grounded language learning in a simulated 3d world, 2017.

[22] Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent, 2020.

[23] Xavier Hinaut, Maxime Petit, Gregoire Pointeau, and Peter Ford Dominey. Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Frontiers in neurorobotics*, 8:16, 2014.

[24] David Yu-Tung Hui, Maxime Chevalier-Boisvert, Dzmitry Bahdanau, and Yoshua Bengio. Babyai 1.1, 2020.

[25] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise?, 2020.

[26] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[28] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, 2018.

[29] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020.

[30] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language, 2019.

[31] Olivier Mangin, David Filliat, Louis Ten Bosch, and Pierre-Yves Oudeyer. Mca-nmf: Multimodal concept acquisition with non-negative matrix factorization. *PloS one*, 10(10):e0140732, 2015.

[32] Alana Marzoev, Samuel Madden, M. Frans Kaashoek, Michael Cafarella, and Jacob Andreas. Unnatural language processing: Bridging the gap between synthetic and natural language data, 2020.

[33] Khanh Nguyen, Dipendra Misra, Robert Schapire, Miro Dudík, and Patrick Shafto. Interactive learning from activity description, 2021.

[34] Rohan Paul, Jacob Arkin, N. Roy, and T. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Robotics: Science and Systems*, 2016.

[35] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[36] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding, 2020.

[37] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning, 2017.

[38] L. Steels. Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21(3):32–38, 2006. doi: 10.1109/MIS.2006.58.

[39] Yuuya Sugita and Jun Tani. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive behavior*, 13(1):33–52, 2005.

[40] Jun Tani. *Exploring robotic minds: actions, symbols, and consciousness as self-organizing dynamic phenomena*. Oxford University Press, 2016.

[41] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, and Hideki Asoh. Symbol emergence in robotics: a survey. *Advanced Robotics*, 30(11-12):706–728, 2016.

[42] Elio Tuci, Tomassino Ferrauto, Arne Zeschel, Gianluca Massera, and Stefano Nolfi. An experiment on behavior generalization and the emergence of linguistic compositionality in evolving robots. *IEEE Transactions on Autonomous Mental Development*, 3(2):176–189, 2011.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[44] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning, 2020.

[45] Honglu Zhou, Asim Kadav, Farley Lai, Alexandru Niculescu-Mizil, Martin Renqiang Min, Mubbasir Kapadia, and Hans Peter Graf. Hopper: Multi-hop transformer for spatiotemporal reasoning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=MaZFq7bJif7`.

[46] Li Zhou and Kevin Small. Inverse reinforcement learning with natural language goals, 2020.

[47] Rolf A. Zwaan and Carol J. Madden. *Embodied Sentence Comprehension*, page 224–245. Cambridge University Press, 2005. doi: 10.1017/CBO9780511499968.010.

6 Supplementary: Temporal Playground Specifications

6.1 Environment

Temporal Playground is a procedurally generated environment consisting of 3 objects and an agent's body. There are 32 types of objects, listed in Fig. 5 along with 5 object categories. Each object has a continuous 2D position, a size, a continuous color code specified by a 3D vector in RGB space, a type specified by a one-hot vector, and a boolean unit specifying whether it is grasped. Note that categories are not encoded in the objects' features. The agent's body has its 2D position in the environment and its gripper state (grasping or non-grasping) as features. The size of the body feature vector is 3 while the object feature vector has a size of 39. This environment is a spatio-temporal extension of the one used in this work [13].

All positions are constrained within $[-1, 1]^2$. The initial position of the agent is $(0, 0)$ while the initial object positions are randomized so that they are not in contact $(d(obj_1, obj_2) > 0.3)$. Object sizes are sampled uniformly in $[0.2, 0.3]$, the size of the agent is 0.05. Objects can be grasped when the agent has nothing in hand, when it is close enough to the object center $(d(\text{agent}, obj) < (size(\text{agent}) + size(obj))/2)$ and the gripper is closed $(1, -1$ when open). When a supply is on an animal or water is on a plant (contact define as distance between object being equal to the mean size of the two objects $d = (size(obj_1) + size(obj_2))/2$), the object will grow over time with a constant growth rate until it reaches the maximum size allowed for objects or until contact is lost.

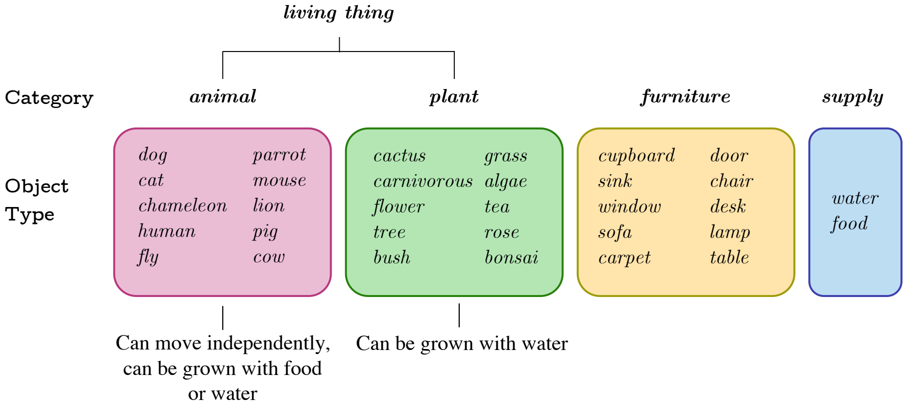


Figure 5: **Representation of possible objects types and categories**. Information about the possible interactions between objects are also given.

6.2 Language

Grammar. The synthetic language we use can be decomposed into two components: the instantaneous grammar and the temporal logic. Both are specified through the BNF given in Figure 6.

Instantaneous grammar:

```
<S>             ::= <pred> <thing_A>
<pred>          ::= grow | grasp | shake
<thing_A>       ::= <thing_B> | <attr> <thing_B> | thing <localizer> |
                    thing <localizer_all>
<localizer>     ::= left of <thing_B> | right of <thing_B> |
                    bottom of <thing_B> | top of <thing_B>
<localizer_all> ::= left most | right most | bottom most | top most
<thing_B>       ::= dog | cat | … | thing
<attr>          ::= blue | green | red
```

Temporal aspect:

```
<S>             ::= was <pred> <thing_A>
<thing_A>       ::= thing was <localizer> | thing was <localizer_all>
```

Figure 6: **BNF of the grammar used in Temporal Playground**. The instantaneous grammar allows generating true sentences about predicates, spatial relations (one-to-one and one to all). These sentences are then processed by the temporal logic to produce the linguistic descriptions of our observations; this step is illustrated in the Temporal Aspect rules. See the main text for information on how these sentences are generated.

Concept Definition. We split the set of all possible descriptions output by our grammar into four conceptual categories according to the rules given in Table 1.

Concept	BNF		Size
1. Basic	<S> ::=	<pred> <thing_A>	152
	<pred> ::=	*grasp*	
	<thing_A> ::=	<thing_B> | <attr> <thing_B	
2. Spatial	<S> ::=	<pred> <thing_A>	156
	<pred> ::=	*grasp*	
	<thing_A> ::=	<*thing* <localizer> | *thing* <localizer_all>	
3. Temporal	<S> ::=	<pred_A> <thing_A> | *was* <pred_B> <thing_A>	648
	<pred_A> ::=	*grow* | *shake*	
	<pred_B> ::=	*grasp* | *grow* | *shake*	
	<thing_A> ::=	<thing_B> | <attr> <thing_B>	
4. Spatio-Temporal	<S> ::=	<pred_A> <thing_A> | *was* <pred_B> <thing_A>	1716
		<pred_C> <thing_C>	
	<pred_A> ::=	*grow* | *shake*	
	<pred_B> ::=	*grasp* | *grow* | *shake*	
	<pred_C> ::=	*grasp*	
	<thing_A> ::=	*thing* <localizer> | *thing* <localizer_all> |	
		thing was <localizer> |	
		thing was <localizer_all> |	
	<thing_C> ::=	*thing was* <localizer> |	
		thing was <localizer_all>	

Table 1: **Concept categories with their associated BNF.** <thing_B>, <attr>, <localizer> and <localizer_all> are given in Fig. 6

7 Supplementary Methods

7.1 Data Generation

Scripted bot. To generate the traces matching the descriptions of our grammar we define a set of scenarii that correspond to sequences of actions required to fulfill the predicates of our grammar, namely *grasp*, *grow* and *shake*. Those scenarii are then conditioned on a boolean that modulates them to obtain a mix of predicates in the present and the past tenses. For instance, if a *grasp* scenario is sampled, there will be a 50% chance that the scenario will end with the object being grasped, leading to a present-tense description; and a 50% chance that the agent releases the object, yielding a past tense description.

Description generation from behavioral traces of the agent. For each time step, the instantaneous grammar generates the set of all true instantaneous sentences using a set of filtering operations similar to the one used in CLEVR [26], without the past predicates and past spatial relations. Then the temporal logic component uses these linguistic traces in the following way: if a given sentence for a predicate is true in a past time step and false in the present time step, the prefix token *'was'* is prepended to the sentence; similarly, if a given spatial relation is observed in a previous time step and unobserved in the present, the prefix token *'was'* is prepended to the spatial relation.

7.2 Input Encoding

We present the input processing in Fig. 7. At each time step t, the body feature vector b_t and the object features vector $o_{i,t}$, $i = 1, 2, 3$ are encoded using two single-layer neural networks whose output are of size h. Similarly, each of the words of the sentence describing the trace (represented as one-hot vectors) is encoded and projected in the dimension of size h. We concatenate to the vector obtained a modality token m that defines if the output belongs to the scene $(1, 0)$ or to the description $(0, 1)$. We then feed the resulting vectors to a positional encoding that modulates the vectors according to the time step in the trace for b_t and $o_{i,t}$, $i = 1, 2, 3$ and according to the position of the word in the description for w_l.

We call the encoded body features $\hat{b}_t$ and it corresponds to $\hat{S}_{0,t}$ of the input tensor of our model (see Fig. 2 in the Main document). Similarly, $\hat{o}_{i,t}$, $i = 1, 2, 3$ are the encoded object features corresponding to $\hat{S}_{i,t}$, $i = 1, 2, 3$. Finally $\hat{w}_l$ are the encoded words and the components of tensor $\hat{W}$.

We call h the hidden size of our models and recall that $|\hat{b}_t| = |\hat{o}_{i,t}| = |\hat{w}_l| = h + 2$. This parameter is varied during the hyper-parameter search.

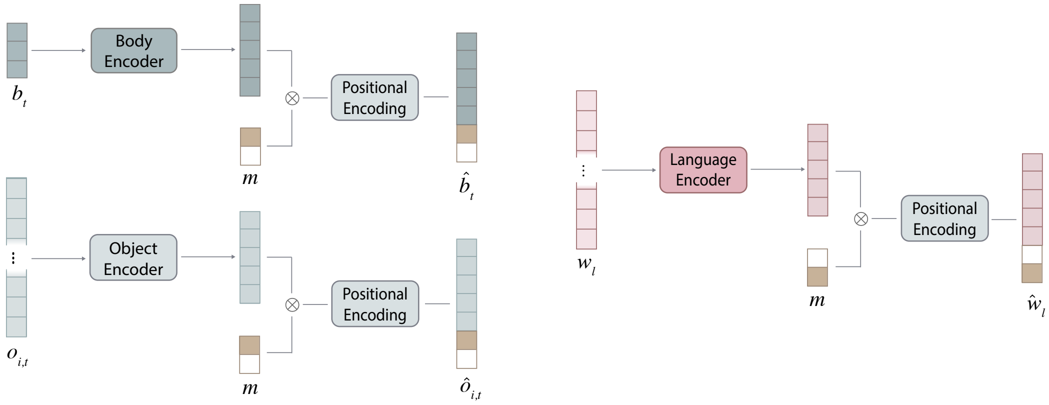


Figure 7: **Diagram representing the projection of the inputs into the same dimension**

7.3 Details on LSTM models

To provide baseline models on our tasks we consider two LSTM variants. They are interesting baselines because they do not perform any relational computation except for relations between inputs at successive time steps. We consider the inputs as they were defined in Section 2.3 of the main paper. We consider two LSTM variants:

1. LSTM-FLAT: This variant has two internal LSTM: one that processes the language and one that processes the scenes as concatenations of all the body and object features. This produces two vectors that are concatenated into one, which is then run through an MLP and a final softmax to produce the final output.
2. LSTM-FACTORED: This variant independently processes the different body and object traces, which have previously been projected to the same dimension using a separate linear projection for the object and for the body. The language is processed by a separate LSTM. These body, object and language vectors are finally concatenated and fed to a final MLP and a softmax to produce the output.

7.4 Details on Training Schedule

Implementation Details. The architectures are trained via backpropagation using the Adam Optimizer[27]. The data is fed to the model in batches of 512 examples for 150 000 steps. We use a modular buffer to sample an important variety of different descriptions in each batch and to impose a ratio of positive samples of 0.1 for each description in each batch.

Model implementations. We used the standard implementations of TransformerEncoderLayer and TransformerEncoder from pytorch version 1.7.1, as well as the default LSTM implementation. For initialization, we also use pytorch defaults.

Hyper-parameter search. To pick the best set of parameters for each of our eight models, we train them on 18 conditions and select the best models. Note that each condition is run for 3 seeds and best models are selected according to their averaged F_1 score on randomly held-out descriptions (15% of the sentences in each category given in Table 1).

Best models. Best models obtained thanks to the parameter search are given in Table 2.

Model	Learning rate	Model hyperparams			
		hidden size	layer count	head count	param count
UT	1e-4	256	4	8	1.3M
UT-WA	1e-5	512	4	8	14.0M
TFT	1e-4	256	4	4	3.5M
TFT-WA	1e-5	512	4	8	20.3M
SFT	1e-4	256	4	4	3.5M
SFT-WA	1e-4	256	2	8	2.7M
LSTM-FLAT	1e-4	512	4	N/A	15.6M
LSTM-FACTORED	1e-4	512	4	N/A	17.6M

Table 2: Hyperparameters for all models

Robustness to hyperparameters For some models, we have observed a lack of robustness to hyperparameters during our search. This translated to models learning to predict all observation-sentence tuples as false since the dataset is imbalanced (the proportion of true samples is 0.1). This behavior was systematically observed with a series of models whose hyperparameters are listed in Table 3. This happens with the biggest models with high learning rates, especially with the -WA variants.

Model	Learning rate	Model hyperparams		
		hidden size	layer count	head count
UT-WA	1e-4	512	4	4
UT-WA	1e-4	512	4	8
SFT	1e-4	512	4	4
SFT-WA	1e-4	512	4	8
SFT-WA	1e-4	512	2	4
SFT-WA	1e-4	512	4	4
TFT	1e-4	512	4	4
TFT-WA	1e-4	512	4	8
TFT-WA	1e-4	512	2	4
TFT-WA	1e-4	512	4	4

Table 3: Models and hyperparameters collapsing into uniform false prediction.

8 Supplementary Results

8.1 Generalization to new observations from known sentences

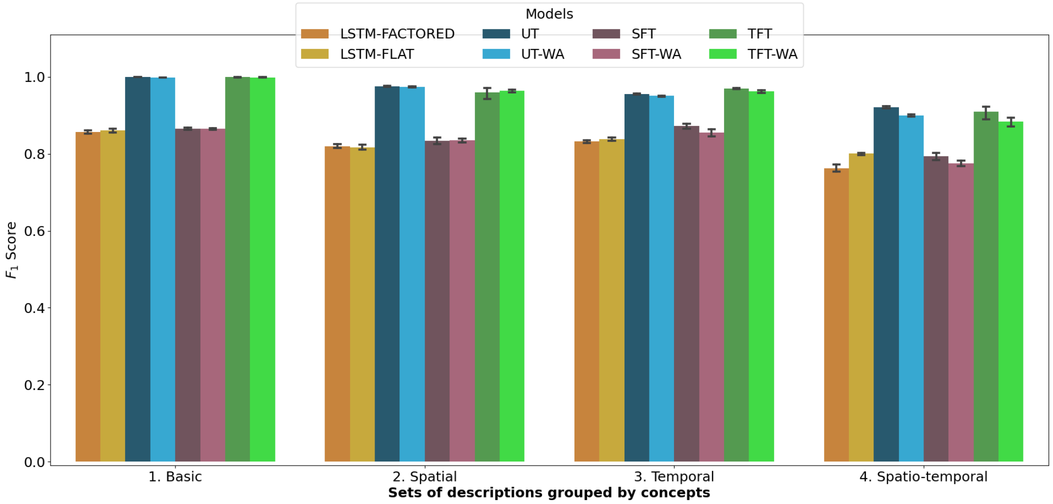


Figure 8: F1 scores of all models on the train sentences with new observations.

In this section we shortly describe an additional evaluation setup we considered. We evaluate the model's f1-scores on sets of sentences that are seen as train sentences, but on newly generated observations. The results are plotted in Figure 8.

8.2 Computing Resources

This work was performed using HPC resources from GENCI-IDRIS (Grant 2020-101594). We used 22k GPU-hours on nvidia-V100 GPUs for the development phase, hyperparameter search, and the main experiments.