Learning Language with Multimodal Models

by

Candace Cheronda Ross

B.S., Howard University (2015)
S.M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by . . . .
Boris Katz
Principal Research Scientist
Computer Science & Artificial Intelligence Lab
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Learning Language with Multimodal Models

by

Candace Cheronda Ross

Title: Learning Language with Multimodal Models

Submitted by: Candace Cheronda Ross

Signature of Author:

Date of Submission:

Expected Date of Completion:

Abstract

Language acquisition by children and machines is remarkable. Yet while children learn from hearing a relatively modest amount of language and by interacting with people and the environment around them, neural language models require far more data and supervision, struggle with generalizing to new domains and overwhelmingly learn from text alone. This thesis explores how knowledge about child language acquisition – particularly the scale and type of linguistic information children receive, how they use feedback, and how they generalize in systematic ways beyond the language input they have been exposed to – can be applied to multimodal language models. In particular, this work focuses on (1) training language models with weak supervision using less data by grounding in vision and (2) exploring the generalization abilities of models in multimodal domains. The first approach trains a semantic parser to map from natural language to logical forms using captioned videos, learning without parse trees or any other annotations. The second approach moves from simply observing videos to a more dynamic setup using a robotic simulator and world states to validate the generated logical forms. These approaches focus on evaluating weak supervision, with training and inference data that are relatively similar; we lastly explore evaluation where the inference data is quite different from training and requires systematic generalizations. One approach tests the role of pretraining and a novel decoding strategy for navigating in a grid world; inference commands and action sequences differ in systematic ways from training. The final approach tests the extent to which pretrained multimodal Transformers models generalize when the demographics in the input images or text differ from their learned social biases.

Thesis Supervisor: Boris Katz
Title: Principal Research Scientist
Computer Science & Artificial Intelligence Lab

This doctoral thesis has been examined by a Committee of the Department of Electrical Engineering and Computer Science as follows:

Principal Research Scientist Boris Katz . . . .
Thesis Supervisor
Department of Electrical Engineering and Computer Science

Research Scientist Andrei Barbu . . . .
Thesis Supervisor
Department of Electrical Engineering and Computer Science

Professor Joshua Tenenbaum . . . .
Thesis Committee
Department of Brain and Cognitive Sciences

Assistant Professor Jacob Andreas . . . .
Thesis Committee
Department of Electrical Engineering and Computer Science

Acknowledgments

First, my committee – to Josh Tenenbaum and Jacob Andreas, I have deep respect for you both as scientists and getting to have you on my committee means a lot to me. I can still vividly remember my first time getting to interact with each of you in a smaller setting – for Jacob, it was right after your CBMM talk when I hung back to introduce myself and for Josh, it was at the Woods Hole summer course at one of the evening social sessions. In both scenarios, I was definitely intimidated to start a conversation and yet in both scenarios, you each were friendly and kind and made me so excited to just chat about science with you. Thank you for contributing to my last chapter at MIT by serving on my committee. To Boris, you are truly one of the kindest people I have ever met and having such a caring advisor is an immeasurably important blessing during grad school. You allowed me explore my interests, celebrated my successes and helped me to navigate through challenges. It's not an understatement to say that every time I meet someone who knows you, one of the first things they say is how thoughtful and kind you are. I emphasize this because your scientific contributions are evident; you also choose to demonstrate such a kind and caring character and I'm very appreciative of that. For my final committee member Andrei, I could likely write an entire section thanking you. I'm deeply grateful to the huge amount of time and energy that you've invested in me and my growth as a researcher, from before I even started grad school to the present day. From before I even started grad school to the present day, you've been instrumental in my growth as a researcher. I can't even count how many evenings you stayed in Stata to help me debug, or to prep slides, or even just to brainstorm about random topics (our office whiteboard reads like a very exciting NLP novel). I feel confident that every person who has worked with you and with Boris would similarly attest to the privilege of that experience.

In particular, a shoutout (with some duplicated names) to the people I've had the pleasure of either sharing an office with over my time at MIT or just seeing each day in Stata – Yevgeni Berzak, Andrei Barbu, Yen-Ling Kuo, Adam Yaari (and Mowgli of course!), Chris Wang, Dylan Sleeper, Alvaro Morales, Matt Staib, Ignacio Cases, Battushig Myanganbayar – and to those I've just generally engaged with either on a project or in Stata. You all have been a wonderful bunch to work near. I'm a huge extrovert and pretty social and I've loved having you all to talk to during our breaks. A bonus thanks to Adam for sharing your Mowgli with us; my success at helping him learn English alongside his native Hebrew is debatable, but I'd say the cuddles and endless treats he got from were definitely a success in his eyes.

To the people in my field who aren't represented in this thesis work formally but are absolutely represented in my growth as a scientist and as a researcher – I'm deeply grateful to have connected with three people in particular from whom I learned a lot of great skills. To Michelle Vanni, who I got the opportunity to work with as a teenager; at the time, I was a teenager a fresh six months out of from the Naval Academy, still learning how to be a civilian. I don't recall if I even knew what NLP stood for when I first arrived. Thank you for welcoming me with open arms, for showing me your excitement about the field and igniting that excitement in me as well. This really laid the foundation for what's been a beautiful time for me in the field. To Mark Yatskar, you were the first person I ever did a reserach interview with and you were the first person I interned with in grad school. Thank you for showing me kindness, for being knowledgeable and patient (I know I had a ton to learn) and for your quintessential hoodie-over-the-head-while-indoors look that few others could rock. Next, to Douwe Kiela, looking back I still remember how intimidated I felt coming to FAIR as an intern and working with you. That intimidation I felt is funny in hindsight because you're a deeply caring person who I believe truly wants the people you work to feel empowered. Thank you for projecting so much confidence onto me as a researcher; I'm grateful for getting to chance to both work with you as a collaborator and connect with you as a friend. I also got support from so many people who aren't in my field but contributed to my growth during my time at MIT. To Mandana Sassanfar, Gloria Anglón, Dean Staton, Leslie Kolodziejski, Janet Fischer, Wes Harris, John Ross Campbell, Kat Howell – people who I So to all of the wonderful researchers mentioned in the paragraphs above, thank you for contributing to my career as a scientist and an extra hug to those of you I've grown to be friends with as well. Being surrounded by people like you all makes the research experience that much more special.

To the brotherhood of Chocolate City – I spent all but my first year of grad school with the privilege serving as your graduate resident advisor. The thought of being a live-in mentor for 30 undergrads felt intimidating before I started but you all welcomed me with open arms. You all were a bonus family on campus and in many ways, you mentored me just as much I mentored you. I'm also super grateful to MIT Gymnastics, ACME and the Black Graduate Student Association as additional communities I got to connect with.

Next up, to my crew outside work! First, to my friends I met through MIT Summer Research Program (MSRP), friends who I've become so close to. Lindsey Backman, Sheena Vasquez and Chad Sauvola, what are the chances we'd meet as undergrads, attend the same grad school and become so close though we came from such different walks of life. In more detail: to Lindsey for being the friend that encourages all of the hilarity behind closed doors; so glad we got to be roommate and GRAs in same building for so long. Your empanadas, black beans and sangria are unforgettable and I look forward to being 80 years old and very willingly waiting days for the meat to season because it's well worth it. To Sheena, for being the friend that channels the most caring vibes everywhere you go. You talk a lot about people's energy and I appreciate this for so many reasons; first, because it's encouraged me to reflect more on this as I meet people, and second, because it's been a solid reminder to me that you're energy is unmatched and I'm so lucky to get to be around that energy every time we hang out. To Chad for being my friend that will randomly walk around West Elm then go get a beer then support me in buying a new duvet or randomly changing the layout of my room; and for being the friend that I can call at 2am when I see a mouse in my apartment (mostly just to drink wine while we avoid the mouse issue).

To Aubrey Dibello and Alex Berry, my best friends from high school and college respectively; how lucky am I that, although we met hundreds of miles from Cambridge, we lived only around a mile away from each other during my PhD. You two were both always there any time I want to just vent, or celebrate exciting news, or just wanted to chat – or when I needed my car started (thanks Aubs)! Aubrey, our running joke that we've known each other since middle school and been friends since high school. I love that our journeys have overlapped so much, from being 8 miles apart in college and then 1 mile apart in grad school; our memories together go way back and there are some that still make me laugh just thinking about them. Beyond just our memories, I really admire your deep deep passions that have led you to some amazing accomplishments. I cannot wait to see you MD-ing it up in New York; your patients will be lucky to have a doctor like you with so much love in his heart. To Alex, my college best friend. I honestly am not even sure how to encapsulate our ridiculous, playful, close friendship beginning from the Naval Academy where we didn't even get to use our first names to now, with both us in Cambridge pursing PhDs at universities ten minutes apart.

To Ashley Hartwell, you're a person I am so so grateful to have become so close to. It's almost funny to look back to our first time meeting each other at lunch in the BSU lounge to being in the same grad orgs, to Martha's Vineyard, randomly sitting in the parking lot during the pandemic, and the many many memories to come, I'm so grateful to have become friends with the funny, caring loving person that you are. Thank you for the wonderful advice (you're always one of the first people I call) and the hilarious laughs; and thank you for being such a thoughtful and brilliant person and for being one of my close friends.

To Azin Saebi, for being so hilarious (my dad literally still talks about how funny you are) while are being so thoughtful. Your impeccable fashion sense, awesome parties (Canadian followed by Iranian trivia night, brunches, kickbacks, and so many others), you have a bright and powerful energy. Here's a shoutout to Beck for giving me one of my besties! To Dominique Wright, for being the friend that brings friends together; you have an indescribable way with people where folks really just gravitate toward you. From the very first trip to San Diego where you called me and were like "yooo book a flight", to LA, to Savannah, to driving from San Francisco to San Diego, to Vegas, to your visits to Boston, to multiple Coachellas, I have so many wonderful memories with you. To Kelly Holden, for being one of the most adventurous people I know. Not many people will just decide they're going on a trip and book a flight, or decide they want to change up their career and start traveling, or decide to pet foster on a whim (yay for Feenie!) but these are things that you do easily. It's a real testament to your courage and how much you enjoy new experiences. (Plus, you save babies for a living so that's extra cool points). I'm also so grateful this brought us so much time together in Boston. To the other people – Zach Nelson, Dmitro Martynowych, Emily Toomey, Marie Shi Feng, Ari Anders, Leilani Gilpin, Kenyon Williams – thank you all for the memories. To Zach in particular for making one of my closest friends so happy. To Dmitro for the delicious eats (the unexpected homemade pandemic bread was very exciting). And to Beck Holden, for being a constant for me for over five years now. There's so much I could say here as well, from our first time meeting in my apartment during that snowstorm, to the many snow days binging Game of Thrones, to our wildly fun whirlwind of a Christmas break where we went from Boston, to Kansas City, to Vegas, to Tampa, to Orlando, to DC and back, to the number of days just laying around ordering food and picking random horror or action films to watch, I could go on and on. I truly believe the most beautiful relationships are those that lead to such fulfilment while being so easy and natural; as I've said so many times so far and I'm sure will continue to say, I am so so indescribably grateful for you.

Now to my loved ones' loved ones; what a privilege I get to love on my friends' families too. To Jill and Anthony Berry, Alex's parents, I always look forward to you coming to visit Cambridge and thank you for inviting me to spend so much time spent on the Vineyard with you. To Lindsey's family, especially her parents Edie and Steve Backman, first thank you for so kindly hosting me when I came to Tampa; The Columbia was delicious and just hanging out downtown and getting to connect you was a beautiful evening. I'm grateful that extended into many other wonderful memories with you (notably the return to The Columbia for the big engagement!) And lastly to Beck's family, Sandra, Mark and Kelly Holden, plus the adorable pets Cane, Feenie and Timmy, I love how you've embraced me and welcomed me in as a bonus family member. Raclette, salsa dancing, Poetry for Neanderthals, Just One, Van Fulls of Nuns – just a small subset of the many wonderful things I attribute to you all. I love having things that remind me of the people I care so much about.

Whew now to my own family! I've been blessed to have a huge extended family, with 16 aunts and uncles, 60+ first cousins, and countless other bonus cousins and loved ones. This makes for an instant support network of people cheering you on and hyping you up-thank you to you all for the endless love. In particular, big old hugs to my cousins Rickkia, Shelby, Alton and Bria; you're the ones closest in age and that I get to see and hang out with the most. See y'all over the years between Disney World trips, huge family get-togethers for birthdays, our upcoming reunion, it's a privilege to have family events like this outside of my daily grad school life. To Bette Lyn, Kenny, Ryan and Chris, for being a second family to me growing up; I love you all so much. To my immediate family – my parents Cherise and Ronald and siblings Camille and Ronald Jr. – I'm so blessed to have such a close, loving and supportive group. I love how silly we are when we're together, how we genuinely have fun and enjoy each other's company, and how, from my earliest memories, I've been raised to believe I could achieve anything. I'm grateful to be more than just siblings with Camille and Ronald Jr. but to also be friends. We enjoy being together, we joke together, I "borrow" Camille's clothes (thank again sis, I promise to return everything!). I could go on and on; instead a quick few words with one of my favorite things about each of you. Dad, for your confidence in the abilities of each of us, which pushes us to achieve so much; Mom, for your kindness and empathy, where I always feel comfortable sharing literally anything with you; Camille, for your drive and desire to be your best that encourages those are you to be their best as well; and Ronald Jr., your deep passion about the things you care about (Sous Chef in just one year?!) that leads you to achieve your dreams.passion and unique. Overall just so much love for y'all. Growing up and thriving in this world as a Black woman can be indescribably difficult at times. My family supported me in being outspoken, in being confident and self-assured, in striving to have empathy and value treating others well. I cannot imagine how different my grad school experience would have been had I not entered with these traits baked into my being. So to my family, friends, mentors, to every person I've mentioned explicitly and implicitly, this thesis is for you!

Contents

Chapter 1

Introduction

1.1 Motivation

Language acquisition is one of the most remarkable aspects of both human and machine learning. Children learn their native language in just their first few years of life through interacting with other speakers and perceiving the world around them. They learn without needing a grammar book or a dictionary and without needing to be corrected when they invariably make a linguistic error (Braine, 1971; Chouinard and Clark, 2003). In natural language processing (NLP), neural language models also impressively acquire powerful linguistic representations through simple training objectives paired with large-scale datasets. Yet while children are robust and have near-universal success at learning their native tongue, language models are brittle and sensitive to training data and hyper-parameters in ways that child learners are not (Pierce, 1992; Rogers et al., 2020; Mosbach et al., 2021). Language models require massive amounts of data (often billions of words) to learn important linguistic concepts (Warstadt et al., 2020; Zhang et al., 2021b). And even with this scale of training data, language models are often detached from perception and only see textual input despite the importance of being grounded in other modalities (Harnad, 2007; Bisk et al., 2020; Lake and Murphy, 2021). Given that children have effectively mastered this complex learning problem, a natural question to ask is – should we incorporate more of what we know about child language acquisition in NLP? While NLP and artificial intelligence in general are not seeking to clone the human learning process, knowledge about how children learn can motivate our work. This thesis considers three focal areas of language acquisition – the type and scale input data, the means of receiving feedback, and the ability to systematically generalize to unseen data – in our work on multimodal language models.

Focus 1: Linguistic & Extra-Linguistic Input

Children learn about 5-10 new words per day and hear around 3-11 million words per year from just 15 months of age (Gleitman and Gleitman, 1992; Hart and Risley, 2003). Language models are trained on orders of magnitude more data. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) for instance have 3 billion and 30 billion tokens in their pretraining datasets, respectively. This number reaches around 130 billion tokens for encoder-decoder models like T5 (Raffel et al., 2020) and continues to increase with new language models being released. Recent work has shown that these large datasets are in fact necessary with the current NLP architectures and learning frameworks for acquiring linguistic generalizations that goes beyond surface-level semantic and syntactic information. Zhang et al. (2021b) find that with 10-100 million words, which is in the range of what children are exposed to during language acquisition, language models can learn representations for syntactic and semantic features like parts of speech and semantic roles but struggle to learn more commonsense or higher level linguistic knowledge. Other work such as Liu et al. (2021) finds that this trend holds even when looking at the performance across training (increasing number of iterations through the data) and across domains. Not only is it the norm to train language models on huge datasets, it is a necessity at present to learn the complex linguistic knowledge necessary for natural language understanding. In Chapter 2 and Chapter 3, we therefore explore approaches that use fewer training examples instead of large-scale datasets.

Extra-linguistic input plays perhaps as important a role as the linguistic input itself. Harnad (2007) explains how our understanding of words and concepts requires perception and cannot be grounded in text alone. Imageability, the ease with which a word can be pictured, plays a key role in early word learning (McDonough et al., 2011). This partly explains why nouns are easier for children to learn than verbs and why the more concrete or “groundable" nouns are learned before more abstract nouns (Gentner, 1982; Gillette et al., 1999). The perceptual tie between what children see and what they hear also means there are acquisition differences for blind children (Bigelow, 1987; Andersen et al., 1993) , again demonstrating that visual perception does play a large role in language learning[1]. Language models absolutely should be grounded and trained with perception (Bisk et al., 2020; Bender and Koller, 2020; Tan and Bansal, 2020). Zhang et al. (2021b) even hypothesize that the lack of grounding partially explains why language models trained with smaller training sets perform poorly on commonsense tasks. While there are many vision and language models (V&L) – VSE++ Faghri et al. (2017), ViLBERT (Lu et al., 2019), VILLA (Gan et al., 2020), ViLT (Kim et al., 2021) to name just a few – these approaches rarely outperform language models trained on text. (Yun et al., 2021) for instance directly compare representations from language-only vs. vision and language and find that the V&L models do not produce significant gains on linguistic tasks even though we know visual information makes a distinct difference for children. Though we do not explicitly explore outperforming language models trained only on text, all of the work in thesis focuses on visually grounded language models.

Focus 2: Feedback & Learning without Corrections

Beyond just differences in the linguistic and extra-linguistic information received, there are also differences in how children receive corrections and update their beliefs about the meaning and structure of the language they are hearing compared to how we train language models. During child language acquisition, children interestingly do not appear to need or even use negative feedback. This means that a child manages to learn that *“Go to bed"* means it’s time for sleep and not time to play, that *broke* is grammatical while *breaked* is not, and many other types of linguistic knowledge, all without being directly corrected by an adult when they invariably make a mistake. Research has even found that children actually tend to ignore linguistic corrections when received. This suggests there is not a practical benefit of direct corrections for their model of learning language (Braine, 1971; Pinker, 1989).

[1] While acquisition patterns differ, the reader should note however blind children do converge learning their native language (Landau et al., 2009).

One theory that supports this finding is that children use *indirect* feedback to validate their guesses about the language they are hearing and to correct errors in the language they are generating. This indirect feedback is often nuanced, and children appear to take a statistical approach to modeling their linguistic beliefs. For instance, adults sometime respond to a child's ungrammatical utterance by rewording it when responding (Chouinard and Clark, 2003). This provides a cue to children that the language they produced may have been erroneous. Regardless of what specifically is used as evidence, children do not require direct feedback and ignore it when given. Indirect information – from other speakers or from the environment – is one observed means of correcting a child's understanding of natural language.

NLP is largely in a pretrain-then-finetune framework, where pretraining uses self-supervision and finetuning uses full supervision (requiring ground-truth, task-specific labels). Full supervision gives the model access to input-output pairs for a given linguistic task – e.g., commonsense questions and answers or sentences and parse trees – which is quite different than the indirect supervision children receive. Full supervision is also expensive, requiring gold labels for examples that are often gathered from human annotators. Adding modalities like vision makes the process even more expensive, explaining in part why V&L datasets tend to be much smaller than language-only datasets. Weak supervision, where indirect information is used as a noisy signal, is much more similar to how children learn, yet is not often used by large models and rarely if ever outperforms supervised methods. In Chapters 2 and 3, in addition to exploring training with a smaller dataset, we also use weak supervision where an executor provides feedback for a training example instead of using the ground-truth answer.

Focus 3: Systematic Generalization & Seeing Beyond Input Data

Having explored the type and scale of linguistic input (the training data) and how feedback is given and used (the level of supervision), the last area to discuss is systematic generalization. We can think about successful systematic generalization where the representations learned for words or concepts are reusable and abstract enough that they can be applied in novel settings. Children learn the word *ball* from a baseball, for instance, and can generalize the concept to novel instances like a basketball, or a soccer ball, or a squeaky toy ball for a pet. Beyond just word acquisition, children (and adults) must also perform higher-level generalizations. One important societal instance is on the basis of demographics, where social biases emerge when humans fail to systematically generalize concepts beyond the racial or gender bounds they have seen. Even if every doctor a person has seen has a been a man, and every reference they've heard spoken about doctors has referred to men, that person still needs to be able to generalize the concept of a doctor to other genders. Blair et al. (2001) show that humans are actually relatively good at generalizing beyond implicit stereotypes; mental imagery – visually picturing evidence that counters a stereotype – is enough to modulate biases. At course and fine-grained levels, children are able to systemically generalize to novel data.

Language models tend to struggle at tasks that require systematic generalization. Many benchmarks and datasets have been developed to explore this very failure in both language and multimodal language models (Gershman and Tenenbaum, 2015; Lake et al., 2017; Bastings et al., 2018; Kim and Linzen, 2020; Ruis et al., 2020). For instance, the SCAN benchmarks that maps natural language commands to action sequences shows that language models struggle to generalize when the commands seen at test time differ largely and systematically from those seen during training (Lake and Baroni, 2018). The COGS benchmark takes a similar approach, using semantic parse trees instead of actions sequences and using more naturalist language (Kim and Linzen, 2020), again showing that language models struggle during inference. This approach has been extended to multimodal models as well, with the grounded SCAN (gSCAN) benchmark using a grid world and requiring an agent to track its world state over time as well; similar to SCAN and COGS, the multimodal models evaluated on gSCAN struggle to generalize during inference. The test time failures are in spite of language models being trained with massive amounts of compute and linguistic data. In Chapter 4, we explore ways to build upon and improve systematic generalization in multimodal settings and in Chapter 5, we present metrics for measuring generalization around demographics, specifically probing social biases in embeddings in multimodal models. While we focus on vision and language models, these metrics for measuring biases can be applied to any model with two distinct modalities.

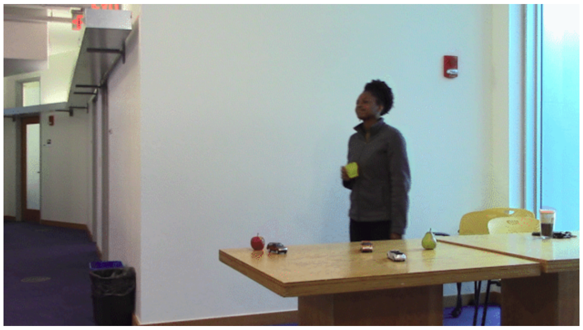
1.2 Research Contributions this Thesis

First in Chapters 2 and 3, we present two weakly supervised, visually grounded semantic parsers that use small-scale training datasets and incorporate extra-linguistic information in the learning process. These chapters contribute to the first and second focus points of using using multimodal, small-scale data and learning without direct feedback. In Chapters 4 and 5, we explore multimodal models' ability to generalize during inference from two angles, which explores the third focal point of systematic generalization. Chapter 4 explores whether flexible, grounded linguistic representations can be learning during pretraining that can help better generalization. Chapter 5 presents metrics for evaluating generalization failures of pretrained vision and language Transformers and exploring when they fail to infer beyond their learned social biases. A detailed description of each approach is provided below.

1.2.1 Weakly Supervised Semantic Parsing

Using Captioned Videos

First, we describe a weakly supervised semantic parser that is grounded in vision, learning from real-world videos paired with English captions. This approach focuses on small-scale training data that does not rely on ground-truth labels during training. The videos depict agents carrying out a series of actions like picking things up, putting them down, passing them to each other, etc; an example of a video from our dataset paired with a caption is shown in Figure 1-1. The parser is trained to map these natural language captions to logical forms, which are formulas representing the syntax and semantics of the sentence. These logical forms are then executed in a visual system called a Sentence Tracker (Siddharth et al., 2014; Yu et al., 2015) that computes the likelihood of the logical form being true conditioned on the video. This likelihood is used as a supervision signal for the parser, directly grounding the process of acquiring linguistic knowledge in vision. The model only has access to the likelihood computed by the tracker and does not use any other annotations such as ground-truth formulas as supervision.



The woman walks by the table with a yellow cup.

$\lambda xyz.$woman x, walk x, near x y, table y, hold x z, yellow z, cup z

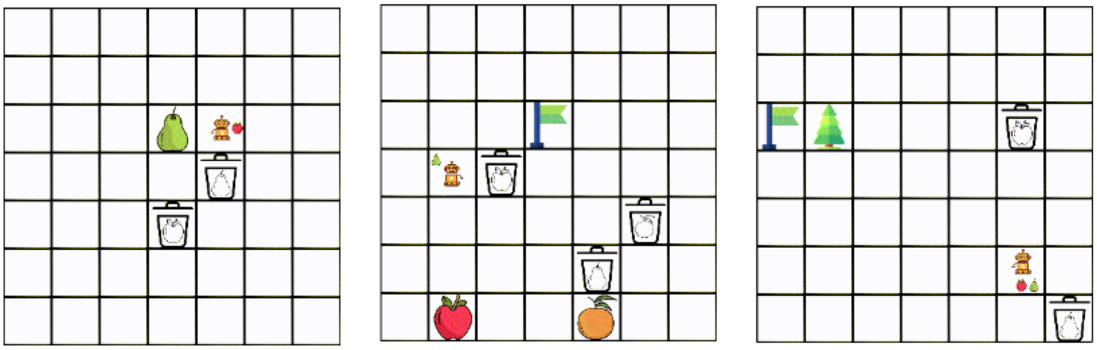
Figure 1-1: An example from our dataset of video-sentence pairs. We show the target logical form, though note that the model is trained without access to these ground truths.

Using a Robotic Executor

For the second approach, we extend the prior work by moving from observing the world through videos to interacting with the world through a robotic simulator. This interactive framework allows us to move from the relatively static language associated with videos to more temporally constrained language. The parser is trained to map English sentences to linear temporal logic (LTL) formulas over finite sequences. The generated LTL formulas are then executed by a pretrained planner adapted from Kuo et al. (2020b); similar to the sentence tracker described above, the output of the planner is used to supervise the parser. These sentences represent complex temporal commands such as *“Wash the apple before bringing it to me"* or *“Go to the staircase avoiding the spill on the floor, or clean up the spill first."* An example from our dataset is shown in Figure 1-2.

1.2.2 Generalizations in Grounded Settings

We next move from exploring how to train models with less data in grounded settings using semantic parsing as a framework to exploring how multimodal models systemically generalize during inference. We consider two different angles. For the first angle, we hypothesize that an encoder-decoder framework that focuses on learning strong, disentangled representations during pretraining along with a shift in generating output during decoding



grammar generated: Eventually, possess the pear and grab the apple and persist.
human annotation: Eventually, hold the apple and take the pear and do this repeatedly.

Figure 1-2: An example from our dataset used to train an LTL-based semantic parser. Each command has three different corresponding environments, and each of these environments is represented by an image above showing the initial state. The commands are shown in their grammar-generated form and in the more linguistically diverse form generated by human annotators on Amazon Mechanical Turk.

can lead to gains in generalization. We use the gSCAN benchmark (Ruis et al., 2020) for this task, which pairs natural language commands and an agent in a grid world with target action sequences.

To our knowledge, no prior work on gSCAN uses any form of pretraining to learn reusable representations. We designed the pretraining objectives to be task-independent, focusing on learning 1) robust visual representations of the grid world, 2) robust linguistic representations for the commands, 3) a mapping between linguistic and visual concepts, and 4) a general spatial and temporal understanding of the agent moving through the grid world. Additionally, we present a new decoding strategy where, instead of predicting an entire action sequence at once or predicting a single action at a time, the model first predicts sub-sequences of tokens and then updates the world state after each predicted sub-sequence. This process repeats until an end of sequence token is predicted. This approach balances the model being able to plan out a trajectory of actions, as is done by the majority of approaches, while also getting intermediate updates of the world state. It is also more efficient than predicting a single token at a time, leading to much faster training.

Another component of generalization difficulties is that language models quickly acquire biases and fail to recover from them in light of new data. We lastly investigate which kind of biases vision and language Transformers learn during their pretraining process. We focus in particular on social biases, which are prejudices about groups of people on the basis of demographic categories like race or gender. To examine this, we selected four powerful pretrained models that differ architecturally and in their pretraining data. We adapt the existing language-only bias metrics of the Word Embedding Association Test (WEAT) (Caliskan et al., 2017) and the Sentence Encoder Association (SEAT) Test (May et al., 2019) to a multimodal space to evaluate these approaches. With Grounded WEAT (GWEAT) and Grounded SEAT (GSEAT), we ask three key questions. 1) *Do multimodal embeddings from pretrained vision and language Transformers contain social biases?* This question seeks to first establish whether embeddings from these pretrained architectures encode social bias and if so, to what extent. 2) *Does counterstereotypical visual evidence that goes against the prejudice impact the encoded bias?* This question seeks to answer what role is played by evidence that goes against a particular stereotype. Humans are robust to changing their views in light of new counterstereotypical evidence, so ideally we would see this in multimodal models as well. 3) *How much of the encoded bias comes from language input versus vision input?* This question seeks to answer how much of the bias measured by the first question comes from the visual input versus the language input. It could be that language completely dominates, or that the visual input is the strongest signal, or that both inputs contribute relatively equally. Better understanding the contributions of each modality can be helpful in deciding the best approaches to mitigating bias. We hope the metrics that answer these questions can collectively help to improve our frameworks and reduce bias over time.

1.3 Thesis Outline

To explore the proposed research problems, this thesis is broken down into the following chapters. In Chapter 2, we present a weakly supervised semantic parser that maps natural language sentences to combinatory categorial grammar parses trained from captioned videos. In Chapter 3, we present a more interactive semantic parser that is paired with a robotic simulator. The input are highly temporal natural language commands that rely on the agent understanding world states, mapped to linear temporal logic. Chapter 4 shows how a pretrained, multimodal encoder-decoder that makes its predictions iteratively leads to improvements in generalization. Chapter 5 presents new metrics, Grounded WEAT and Grounded SEAT, for probing social biases in multimodal embeddings. Lastly, Chapter 6 summarizes the contributions of each chapter and how they tie into our overall interests in using theories from child language acquisition as inspiration and motivation for approaches in NLP.

Chapter 2

Semantic Parsing as a Model of Language Acquisition using Captioned Videos

Children learn language in naturalistic environments from receiving a relatively small amount of linguistic input and without direct corrections. In this spirit, we present a weakly supervised semantic parser trained on a dataset of captioned videos without any other annotations or labels. The semantic parser learns a lexicon that maps tokens to syntactic and semantic types, and then uses these types to build parse trees. The logical forms, which are the root of the parse tree, are then executed by a visual system that computes the likelihood of the parse being true given the video. With this supervision alone, the semantic parser can successfully learn to map sentences to logical forms. It does so despite the ambiguity inherent in vision, where a sentence may refer to any combination of objects, object properties, relations or actions taken by an agent in a video. We also released the dataset of sentences and videos paired with logical forms for training and evaluation along with the code for reproducability.

2.1 Motivation

Children learn language from observations that are very different in nature from what semantic parsers are trained on. Rather than receiving direct feedback such as the part of speech associated with the words they are hearing or corrections about incorrect grammatical interpretations, children use interactions with speakers and their environment to update their beliefs about the language they are hearing. This is a particularly impressive feat – children learn the structure of the language they hear without ever seeing that structure overtly. This weak and indirect supervision where most of the information is obtained through passive observation poses a difficult disambiguation problem for learners: how do you know what the speaker is referring to in the environment, i.e., what does the speaker mean? Speakers can refer to actions, objects, the properties of actions and objects, relationships between those actions and objects and other features in the environment and generally combine multiple features into complex sentences. Moreover, speakers need not refer to the most visually salient parts of a visual scene. Here, we induce a semantic parser by simultaneously resolving visual ambiguities and grounding the semantics of language using a corpus of sentences paired with videos without other annotations.

The goal of semantic parsing is to convert a natural language sentence into a representation that encodes the sentence's meaning and structure. These representations called logical forms – expressed in lambda-calculus in our case – can be used for a variety of tasks such as querying databases, understanding references in images and videos, and answering questions. To train the parser presented here, we collected a dataset of real-world videos paired with English captions generated from annotators on Mechanical Turk. We balanced the dataset such that the raw statistics of the co-occurrences of objects and events are not informative. Given these caption/video pairs as input, the parser hypothesizes potential meanings for the captions as logical forms in lambda-calculus. We then used a modular vision system that constructs a specific detector for each hypothesized logical form and then determines the likelihood of the logical form being true of the video. This likelihood is used as supervision for the parser. This approach is inspired by a child forming a hypothesis about the meaning of language they're hearing and then seeing how well this hypothesis fits their understanding of the objects and how they interact in the world. To evaluate our trained parser, we manually annotated each caption with its ground-truth semantic parse. These annotated parses are for evaluation only and are never used to train our model.

At training time, we jointly learn using both the semantic parser and the vision system; at evaluation, we use the learned parser with the language input only and do not use or need the videos. The semantic parser learns a set of weights θ and a lexicon Λ. For both the parser and the associated language-vision component, the lexicon Λ is used to generate and validate parses. To create new lexical entries, we employ a variant of *GENLEX* from Artzi and Zettlemoyer (2013) that takes as input a validation function — in our case, the validation function is the compatibility between a generated parse and the video. *GENLEX* uses an ontology of predicates, a validation function, and templates from the current lexicon to construct new syntactic and semantic forms.

The grounded semantic parser must learn these parameters despite three sources of difficulty and noise. First, the vision system is the sole source of validation for whether a parse is valid, but this component may fail because computer vision is far from perfect. Notably, this component works in part by using bounding boxes from an object detector run over frames of the video to compute the compatibility with the predicted parse. Many objects in the video are small, partially occluded, and blurry given that we are taking still frames from videos. Overcoming this noise in the process requires large beam widths and a low confidence threshold for the detector to avoid falling into local minima due to these errors.

Second, an infinite number of possibly-erroneous parses are true of a video. When children learn language, they face this same challenge as they do not have access to bounding boxes or to logical forms. The parse $\lambda x.\mathit{person}(x)$ as well as many other seemingly reasonable parses are true for most videos and will therefore receive a high likelihood score from the vision system. This and similar parses cannot be distinguished from the ground-truth parse — which is not available — by the vision component. Our approach is a less constrained environment than many other approaches to semantic parsing because our validation function may be true when it should not be. Additionally, this leads to *GENLEX* creating many special-purpose entries for words that just happen to fit the peculiarities of any video. Here, we find two different problems: 1) the assigning of empty semantics to many words since the likelihood of a subset of a parse is always the same or higher than the whole parse and 2) excessive polysemy where the meaning of a word is highly specific to some irrelevant feature in a video. We therefore introduce two features to the parser inspired by the Rational Speech Acts model (Frank and Goodman, 2012) that bias it against empty semantics and against excessive polysemy when generating and selecting lexical entries for the derivation.

A third difficulty is that computer vision models can be computationally expensive and we need to run many evaluations of parse-video pairs to train the parser. To overcome this, we construct a provably-correct cache that keeps track of failing sub-expressions. This is possible because of a feature of this particular vision-language scoring function: the score decreases monotonically with the number of constraints. Therefore, if any sub-expression of a parse has been generated before for a given video and rejected, we know this parse can be rejected as well. With these improvements, the grounded parser learns to map sentences into semantic parses despite facing a challenging setting with limited examples and much ambiguity.

This work makes several contributions. First, we show how to construct a semantic parser that learns to represent the meaning and structure of language in a setting closer to that of children by grounding in perception and using a modest amount of data. Additionally, we demonstrate how to jointly resolve linguistic and visual ambiguities at training time in a way that can be adapted to other semantic parsing approaches. The visual input plays a crucial role in disambiguating sentences with multiple semantic interpretations; in many cases, the true semantic meaning being conveyed by the speaker cannot be discerned without this additional context. Next, we demonstrate how this and similar approaches can be bootstrapped by using a small number of annotated sentences combined with a large number of videos paired with unannotated sentences in order to improve performance. Annotating just a handful of examples by hand serves as a helpful starting point for the model. Lastly, we release a clean dataset systematically constructed and annotated on Mechanical Turk for joint visual and linguistic learning tasks.

2.2 Related Work

2.2.1 Rule-based & Fully Supervised Approaches

Early approaches to semantic parsing were largely rule-based (Winograd, 1971; Woods, 1973; Johnson, 1984). These rule-based approaches had the benefit of being quite interpretable, yet suffered from being brittle as these rules were domain-specific and required hand annotation. Later approaches moved from rule-based to statistical systems, using fully supervised learning algorithms by training with pairs of sentences and meaning representations (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005, 2007; Kwiatkowksi et al., 2010; Andreas et al., 2013). Zettlemoyer and Collins (2005) present a probabilistic approach to inducing a grammar for generating logical forms alongside a probabilistic model for ranking the generations. Zettlemoyer and Collins (2007) relax the previous approach by introducing new CCG combinators. These new combinators allow for more flexible word order, e.g., *flights one-way* instead of *one-way flights*, reducing the rigidity of typical CCG-based lexicons. Kwiatkowski et al. (2011) introduce factored lexicons, where lexical entries are broken into lexemes, that represent the semantic meaning, and templates, that represent the syntactic meaning. Approaches using factored lexicons learn with less data and are better at learning to map to words that appear in polysemous contexts.

Neural approaches to semantic parsing alleviated the need to formally define rules or the lexicon, often adapting approaches from neural machine translation. Dong and Lapata (2016) present two variants of an LSTM-based encoder decoder, one with a hierarchical tree decoder and another that uses soft attention to align the natural language and parse. Jia and Liang (2016) take a data recombination approach, where they first induce a context-free grammar to map sentences to newly generated samples, then train a RNN to map the samples to semantic parses. They additionally employ an attention-based copying mechanism that allows words from the input sentence to be directly copied to the parse.

2.2.2 Weakly Supervised Approaches

Many prior approaches to semantic parsing have removed the need for labeled parses by training in a weakly supervised setting. Matuszek et al. (2012) train a semantic parser for an object selection task. Given an image of differently colored and shaped objects, the parser generates logical forms that select the referenced objects described by the sentence. Our work can be thought of as a variant of the object selection task where the objects and agents are dynamic and an understanding of actions and interactions is necessary. This represents one key difference of moving from static images to videos. Another approach is that of Berant et al. (2013) who train a semantic parser by execution, in this instance using question-answer pairs from the knowledge base Freebase. The questions are parsed to formulas that are executed in Freebase and the feedback comes from whether the correct answer is returned. Databases have far less ambiguity than videos; every question has (usually) a single correct answer and that answer will be deterministically returned by the database producing a binary response about the parse. Our model is noisy, producing likelihoods that are more difficult to interpret than a single answer from a database. Additionally, databases do not have a notion of perception.

In an approach grounded in a robotic simulator, Artzi and Zettlemoyer (2013) train a weakly supervised semantic parser that maps robotic commands to executable formulas. The validation function compares the world state after execution to the goal state. The simulator, much like knowledge bases, is noiseless and deterministic; this makes for a smoother validation function than our model's validation, which relies on fallible perception. In a slightly different approach, Berant and Liang (2014) learn to parse sentences using a paraphrase model where candidate parses are mapped into canonical representations and then paraphrased to choose the one that best matches the input sentence. One might consider our approach concerned with visual and not just linguistic paraphrases, where the linguistic representation best “paraphrased" by the video is the one selected by the model.

Learning to understand language in a multimodal environment is a well-developed task in other frameworks outside of semantic parsing as well. Siddharth et al. (2014) and Yu et al. (2015) focus on acquiring the meaning of a lexicon from videos paired with sentences. Their approach differs from our work in that they assume a fully-trained parser. Our approach instead focuses on both inducing the lexicon and learning the parameters of a parser. Visual question answering (VQA) is another similar area where models are trained on datasets like Antol et al. (2015) to understand complex visual scenes alongside language. The goal of our work is not to produce answers for any one set of questions, although this should be possible with our approach. We instead learn to predict the structure of the sentences and their meaning. This is a more general and difficult problem, in particular because at test time we do not receive any visual input, only the sentence. The resulting approach is reusable, generic and more similar to the kind of general-purpose linguistic knowledge that humans have. For example, one could use it to guide robotic actions.

Other works in understanding language using vision are that of Wang et al. (2016), who create a language game to learn a parser in a block world. This approach, similar to Artzi and Zettlemoyer (2013), does not contain the noisy, fallible perception. Al-Omari et al. (2017) acquire a grammar for a fragment of English and Arabic from videos paired with sentences. They learn a small number of grammar rules for a language restricted to robotic commands. Learning occurs mostly in simulation and with little visual ambiguity, and the resulting model is not a parser but a means of associating *n*-grams with visual concepts. This is a non-exhaustive overview of some approaches to weakly supervised semantic parsing and the work being done in visually grounded language models.

2.3 Task

Given a dataset of captioned videos, $D = \{(s, v)\}$, where *s* is the caption of video *v*, we train the parameters and lexicon, θ and Λ, of a semantic parser. At training time, we perform gradient descent over the parameters θ and employ *GENLEX* (Zettlemoyer and Collins, 2005) to augment the lexicon Λ. The objective function of the semantic parser is written in terms of a visual-linguistic compatibility between a generated parse p with corresponding logical form $\boldsymbol{z}$ and video v. Recall that logical forms are the root of the parse tree. This compatibility computes the likelihood of the logical form being true of the video, $P(v|\boldsymbol{z})$. At test time, we take as input a sentence without an associated video and produce a semantic parse. We could in principle also take as input the video and produce a targeted parse for that visual scenario. This is a problem similar to that considered by Berzak et al. (2015), but we do not do so here.

2.4 Model

2.4.1 Semantic Parser

We adopt a semantic parsing framework similar to that of Artzi and Zettlemoyer (2013) using the Combinatory Categorial Grammar (CCG) formalism (Steedman, 1996, 2000), although the general approach of using vision as weak supervision for semantic parsing generalizes to other parsers as well. CCG is a powerful formalism that couples syntax and semantics and represents language compositionally. CCG-based semantic parsing works by learning a lexicon that maps tokens to syntactic and semantic types; a small number of fixed unary and binary derivation rules then use the lexicon to build parse trees. These rules we include are the forward and backward application define the direction, either right or left for forward and backward application respectively, and type-raising (Carpenter, 1997), which allows for a given syntactic type mapped to a token "raise" to a different type contextually. So concretely, the parser takes as input a sentence as sequence of tokens and a (learned) lexicon of potential syntactic and semantic types for each token and uses these types to step through the derivation of a parse tree. Each step of the derivation works by using the combinatory rules to combine the syntactic types; each time two syntactic types are combined, the corresponding semantic types are compositionally combined as well. The parser accepts a derivation when the tree reaches a single node; this single node represents the logical form. Figure 2-2 shows a parse starting with tokens and their syntactic types along with each rule being applied. Additionally, we use the following type system for our semantic representations:

- *e:* entity; this can refer to any constant in our system
- *p:* people
- *o:* inanimate objects

- *a:* actions such as *pick up* and *drop*
- *t:* truth-value

where *p*, *o* and *a* are new types differing from previous approaches.

Concretely, consider a case from Figure 2-2 where a determiner is attached to a noun, *the cup*. The tokens the and cup are hypothesized to have syntactic types NP/N and N (a function returning NP given an argument on the right side and a noun) and semantic type $\lambda fx.fx$ and $\lambda x.\text{cup}(x)$ (the identity function and a function that adds a cup constraint). These two derivations can be reduced by forward application, denoted by $>$. Both the syntactic and semantic types are applied and reduced, which means the semantics helps guide the syntax. Derivations that produce illegal operations, such as applying an argument to a constant, are forbidden.

This process produces multiple hypotheses so following Zettlemoyer and Collins (2005) and Curran et al. (2007), we adopt a weighted linear semantic parser. For each sentence paired with its hypothesized derivation, this approach computes a feature vector ϕ and a parameter vector θ. Given a sentence s, a parse p, a lexicon Λ, the set of all possible parses for that sentence with that lexicon, $P(s, \Lambda)$, and an n-dimensional feature vector computed for that sentence and parse, $\phi(s, p)$, the parser finds the best parse p^* by optimizing:

$$p^* = \operatorname*{argmax}_{p \in P} \ \theta \cdot \phi(s, p) \tag{2.1}$$

Using a fixed-width beam search, the parser enumerates derivations by choosing a potential syntactic and semantic type for each token from the lexicon and choosing a set of derivation rules to apply. For the i-th training sample d_i, consisting of a sentence s_i and a video v_i in dataset D and the feature function, the parser maps sentence s_i to the set of parses E. Using the validation function to determine valid and invalid predictions, the set of margin-violating positive, E^+, and negative, E^-, parses are used to update the parameters θ according to:

$$\theta \rightarrow \theta + \frac{1}{|E_i^+|} \sum_{e \in E_i^+} \phi_i(e, v_i) - \frac{1}{|E_i^-|} \sum_{e \in E_i^-} \phi_i(e, v_i) \tag{2.2}$$

. After each sweep through the dataset, the lexicon Λ is augmented using the modified *GENLEX* from Artzi and Zettlemoyer (2013), which does not require the ground-truth logical form. At no point is the logical form needed for updating the lexicon or parameters; we rely instead on a visual validation function to compute the margin-violating examples.

To further encourage the model to produce logical forms that convey semantic meaning and do more than just satisfy the vision system, we introduce two new features. Models of communication such as the Rational Speech Acts model (Frank and Goodman, 2012) predict that speakers will avoid inserting meaningless words. Using this as motivation, one feature counts the number of predicates mapped onto semantic forms which are empty that occur in each parse. The other feature attempts to prevent excessive polysemy by counting how many new semantic forms are introduced for existing tokens by the generated entries from each parse. As the parser becomes more capable of handling sentences in the training set, these features begin to bias it against adding empty semantics and new semantic forms.

Rather than attempting to learn a fixed lexicon that directly maps tokens to semantic and syntactic parses, we use a factored lexicon like that of Kwiatkowski et al. (2011). This represents tokens and any associated constants separately from potential syntactic and semantic types. For example, the token *chair* is associated with a single constant $\texttt{chair}$; *chair* $\vdash [\texttt{chair}]$. In addition to the token-constants pairs, there exists a list of pairs of syntactic and semantic types along with placeholders for constants; in the case for *chair*, a useful type might be $\lambda v.[N : \lambda x.\texttt{placeholder}(x)]$. When parsing, each token is applied to a potential syntactic and semantic type and the derivation proceeds from there. The factored lexical entries allow for far greater reuse; the model learns a small number of constants that a word can imply separately from a small number of syntactic and semantic types for any word. The lexicon is seeded with a small number of generic combinations of syntactic and semantic types. We sampled a small number of captions (less than 1% of total examples) and manually parse them; the components of the parse are used to populate the seed lexicon. These sentences were excluded from both the training and test sets and were only used to populate the seed. There are 100 unique lexical entries in the seed; other similar grounded approaches have a similar number of seed lexical entries (e.g., Artzi and Zettlemoyer (2013) provide 141 possible types).

In summary, we learn a weighted linear CCG-based parser searches over potential lexical entries, applying the token to different syntactic and semantic types and over multiple hypotheses for which rule should be applied. At training time, in order to learn a reasonable lexicon and set of parameters, a supervision signal is required to validate candidates. We provide that supervision using the vision system described below in Section 2.4.2.

2.4.2 Sentence Tracking

To score a predicted logical form given a video, we employ a framework similar to that of Yu et al. (2015). This approach constructs a model specific to the logical form by extracting the number of participants in the scene described by a caption as well as the relationships and properties of those participants. It builds a graphical model with components for each participant and the relationships between them. First, each participant is localized by an object tracker. Next, each relationship is encoded by Hidden Markov models (HMMs) that model the temporal constraints of the relationships between the Viterbi-based trackers for the participants. Recall that each logical form in the CCG formalism is a lambda expression with a set of binders, whose domain are objects, and a conjunction of constraints that refer to these binders. In essence, this notes which objects should be present in a scene (participants) and what static and changing properties and relationships (temporal constraints) those objects should have with respect to one another.

To make this work, first we use an object detector to generate a large number of bounding boxes for the frames of the video. As noted earlier, we use a low confidence threshold for the object detector erring on the side of over-detecting instead under-detecting participants. To detect the objects in each frame of the video, we use two off-the-shelf object detectors, OpenPose (Cao et al., 2017; Simon et al., 2017; Wei et al., 2016) for people and YoLo (Redmon and Farhadi, 2018) for the remaining objects. We ran a few fine-tuning iterations on YoLo for the specific objects in our video that did not overlap with YoLo. We wanted to avoid overfitting the object detector to our specific dataset while also recognizing the importance of high-quality detections for the Sentence Tracker.

The object trackers weave these bounding boxes into high-scoring object tracks and use constraints to verify if the tracks have the desired properties and relations. Object trackers are a maximum-entropy Markov model with a per-frame score f, the likelihood that any one object detection is true, as well as a motion-coherence score g, the likelihood that the bounding boxes selected between frames refer to the same object instance. Given a logical form $\boldsymbol{z}$ with L participants and a video v of length T, Equation (2.3) shows the optimization where J is a set of L candidate tracks ranging over every hypothesis from the object detector and b is a candidate object detection.

$$\max_{J} \sum_{l=1}^{L} \left(\sum_{t=1}^{T} f(b^{t}_{j^{t}_{l}}) + \sum_{t=1}^{T} g(b^{t-1}_{j^{t-1}_{l}}, b^{t}_{j^{t}_{l}}) \right) \tag{2.3}$$

Next, the Sentence Tracker must determine if an object track is following the set of behaviors implied by the logical form by using a collection of HMMs. Each HMM has a per-frame score h that observes one or more objects tracks (depending on the number of participants in the behavior being modeled) and a transition function a that determines the temporal sequence of the behavior. Given a video v of T frames and logical form $\boldsymbol{z}$ with C behaviors, also termed constraints, Equation (2.4) shows the optimization where K is a set of states, one for each constraint, and γ is a linking function.

$$\max_{J,K} \sum_{c=1}^{C} \left(\sum_{t=1}^{T} h_c(b^{t-1}_{j^{t-1}_{\gamma^1_c}}, b^{t}_{j^{t}_{\gamma^2_c}}, k^{t}_{c}) + \sum_{t=1}^{T} a_c(k^{t-1}_{c}, k^{t}_{c}) \right) \tag{2.4}$$

For clarity, we find the global optimum for a linear combination of Equation (2.3) and Equation (2.4). The Sentence Tracker computes the likelihood that the logical form is true of the video by jointly optimizing over the sum of Equations (2.3) and (2.4) where the linking function γ connecting the two equations. The Viterbi algorithm carries out this optimization in time linear in the length of the video and quadratic in the number of detections per frame. The result is a likelihood of the logical form being true of a video. This is used to create the visually grounded model that supervises the parser with vision. The tracker can also produce a time series of bounding boxes that make explicit the groundings of the sentences, though we do not use these directly here. Inference proceeds jointly between vision and the logical form where the computed likelihood and gradient update steps guide the parser to focus on events and properties that might otherwise be missed.

2.5 Dataset

We collected and annotated a dataset of captioned videos alongside logical forms generated by manually parsing the caption. The videos contain either one person or a group of people carrying out one of 15 actions, such as *pick up*, *put down*, *drop*, with one of 20 objects spanning 10 different colors. We control for 11 spatial relations between objects and actors. Videos were filmed in multiple locations with multiple agents but care was taken to ensure that the background and agents are not informative of the events depicted.

After filming the videos, we used Mechanical Turk to generate the captions. We asked participants to provide sentences that describe something about the video. We intentionally did not specify what participants should describe to avoid biasing them and to add richness to the dataset. This did occasionally lead to sentences that referred to properties of the video that are well beyond the capacities of the vision system, e.g., descriptions of an agent being lazy or references to the camera's movement. To ensure high-quality examples, two trained annotators manually read through and flagged sentences that were either ungrammatical, contained misspellings or outside of the scope of the vision system. Examples of excluded sentences are shown in Table 2.1.

Grammatical Errors
One man is walking on the towards to another one.
A man holds a yellow chair at chest level was he walks towards a second man.
A guy in striped shirt cross across the room.
Another man is keep the green color bag on the floor.

Spelling Errors
Two men life the chairs at the same time.
One man is hodling green bag.
Both are wearing switers.
Two men walk up to a man in a plad shirt.

Outside of Vision Scope
She holds up the toy car and looks into the camera.
The man with no book bags is lazy and making his friend hold both.
A man who knows he is being stared at moves his bag to his other hand.
A man works out his right arm.

Table 2.1: These are a subset of sentences that were excluded from the final dataset, sorted by the errors they contain. Some excluded sentences contain multiple errors.

Two trained annotators additionally annotated each sentence with a logical form using a set of 34 predicates. Recall that in our weakly supervised framework, the parser does not receive these annotations; the ground-truth logical forms are solely used for evaluation and for supervised benchmarks for comparison. Each logical form was additionally reviewed and corrected by one other annotator. In total, the dataset contains 1200 captions from 401 videos. This is comparable to the size of other datasets used for semantic parsing such as two datasets from Tang and Mooney (2001) with 880 and 640 examples respectively and the navigation instruction dataset (Chen and Mooney, 2011) with 706 examples (containing 3236 single sentences). The sentences comprising our dataset contain 169 unique tokens with an average of 7.93 tokens per caption. There are an average of 2.31 objects per caption.

2.6 Evaluation

2.6.1 Experimental Setup

We adapted the Cornell Semantic Parsing Framework (Artzi, 2016) to train and evaluate the parser. The training, validation and test sets are 720, 120 and 360 examples respectively. These fixed splits were used in all experiments. During training, each hypothesized parse for each sentence is marked as either correct or incorrect, using either direct supervision with the target parse or compatibility with the video, depending on the experiment. We use beams of 80 for the CKY-parser and *GENLEX*. CCG-based semantic parsers are seeded with a small number of generic combinations of syntactic and semantic types. We provide 98 seed lexical entries; to compare relative sizes, the weakly supervised approach of Artzi (2016) seed with 141 lexical entries. The full seed lexicon is shown in Appendix A.

2.6.2 Results

Figures 2-3 and 2-4 summarize the experiments and ablation studies performed. The metrics we use when reporting results are *exact match*, where the predicted logical forms must perfectly match the target logical forms, and *near miss*, where a single predicate in the predicted logical form is allowed to differ from the target. Experiments were averaged across 5 runs.

First as a baseline, we directly supervised the parser with the target logical forms. The model achieved high performance as we exptected with F1 scores of 0.841 and 0.911 for the exact match and near miss cases. This provides a helpful upper bound for comparing the performance under weak supervision. We also evaluate the performance of direct supervision as a function of training set size, shown in Figure 2-4. Performance expectedly decays slightly as we use less training data; this provides a helpful metric for how much data we should use if we want to bootstrap parsers in the future under direct supervision and it provides a comparison for around how much supervised data in the framework produces similar results to the perceptually grounded data.

Next, to establish chance-level performance, we trained the directly supervised approach on shuffled labels, where we assigning random logical forms to each sentence. This is more powerful than a simple chance-level performance calculation as the parser can still take advantage of any dataset biases. Even with the ability to exploit potential biases, performance is very low with F1 scores of 0.136 and 0.349 for the exact and near miss metrics. For our last baseline, we added noise to the directly supervised parser to simulate the unreliable nature of vision and, to an extent, the ambiguities inherent in vision. A certain percentage of the time, the compatibility function returned true or false randomly when given a hypothesized logical form. Figure 2-4 shows performance of the noisy baseline as a function of how much noise was introduced.

For the weakly supervised semantic parser, we train by parsing sentences describing videos to logical forms and computing the compatibility between the predicted logical form and the video. This approach produced 0.2 and 0.6 F1 scores for the exact and near miss metrics. This is far beyond chance performance and corresponds to direct supervision with around 55% noise (recall noisy performance here was 0.22 and 0.39 F1 for the exact match and near miss cases). There are a number of reasons for why performance is not perfect and is much lower than direct supervision with the full dataset. First, the evaluation metrics cannot consider equivalences in meaning, just form. A hypothesized parse may carry the same meaning as the target logical form yet it will be considered incorrect. This is less of a problem with direct supervision where the preferences that annotators have for a particular way of encoding the meaning of a sentence can be easily learned by observing the target forms. In the grounded case, this cannot be learned; visually equivalent parses are equally likely. This is just one difficulty with using structured representations and unfortunately exists in most formalisms used for parsing.

A second challenge that we've noted before is that computer vision is unreliable, i.e., object detectors fail. We find that in many of our videos while person detection is fairly reliable, object detection is unreliable. The final challenge that makes this problem difficult is that vision in the real world is very ambiguous. Predicates like *hold* are true in almost every interaction in our videos. This makes learning the meanings of words much more difficult resulting in the grounded parser often adding useless entries into the predicted logical forms or substituted one predicate for a similar one. The near miss metric shows that overall the parser learned reasonable logical forms. Figure 2-5 shows six examples from our dataset along with expected and predicted parses. We note that while this approach is a challenging setup, children do manage to learn and disambiguate word meaning by observing the environment. And in spite of these difficulties, the parser still manages to learn a lexicon encoding the meaning and structure of the sentences it was trained on.

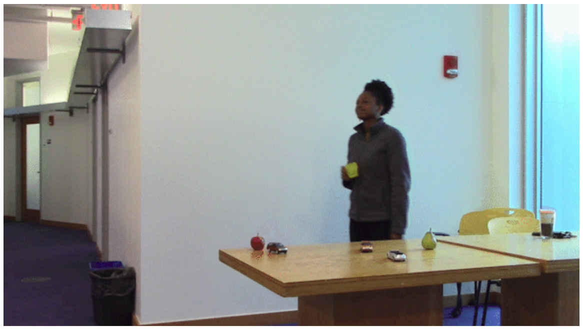
Lastly, we ran an ablation to understand how much of the performance of the grounded parser comes from visual correlations, like the presence or absence of particular objects, as opposed to the more complex and cognitively relevant spatio-temporal relations like actions. To do so, we removed the reliance on all visual features besides objects; this means actions and spatial relationships between participants are ignored. The resulting grounded parser for this ablation accepts *any* hypothesized parse as long as the objects mentioned in that parse are present in the video. This led to a significant performance drop, near-chance level performance on the exact metric, F1 0.05, and nearly half the F1 score on the near miss metric, 0.37. This demonstrated that having a sophisticated vision system to infer about agents and interactions is crucial for learning in this framework.

2.7 Discussion

We present a semantic parser that learns a lexicon mapping words to representations of their meaning and structure using weak supervision grounded in perception. During inference, the model parses sentences without the need for visual input. Learning by passive observation in this way extends the capabilities of semantic parsers and points the way to a more cognitively plausible model of child language acquisition. Nonetheless, several limits still remain. Evaluation metrics do not always capture how well predictions actually match targets. This is because measures of correctness depend on a match to a human-annotated logical form; this is an overly strict criterion that also plagues fully-supervised syntactic parsing (Berzak et al., 2016). Since two different logical forms may still express the same meaning, it is not yet clear what an effective evaluation metric is for these scenarios. Perhaps other formalisms that focus on easily computing equivalence while still providing the combinatory linguistic power of CCG will be helpful. In addition, learning in such a passive scenario by observing videos is hard as correlations between events can be very difficult to disentangle. For instance, e.g., every *pick up* event involves a *touch* event and it is not immediately apparent which event is being referenced just by looking at the video. A more sophisticated action recognition system might be useful, but the constraints are that the system would still need to be computationally efficient.

Looking at the experimental results, an interesting source of error comes from visual ambiguities from the object detector. At the level of relative motions of labeled bounding boxes, the analysis performed by our vision system has difficulty distinguishing certain parts of actions. For example, carrying a shirt and wearing a shirt appear very similar to one another as they are actions that mostly involve moving alongside a person detection. Moreover, since every agent is wearing a shirt it becomes more difficult to learn to distinguish the two actions using positive evidence alone, i.e., a maximum likelihood approach. A more robust vision system, perhaps including object segmentations, person pose, and weak negative evidence for the occurrence of actions, would likely significantly improve the results presented.

In the future, we intend to add a generative model along with a physical simulation allowing the learner to imagine scenarios where a predicate might not hold. This would help mitigate systematic correlations between sentences and videos. The sentences selected here were all chosen such that they are true of the video being shown, yet much of what people discuss is ungrounded, or at least not grounded in the current visual scene. We additionally intend to combine the weakly supervised parser with an ungrounded parser and learn to determine whether a sentence should be grounded visually during training. This would allow the model to handle sentences that do not refer to visual referents.



Sentence: *The woman walks by the table with a yellow cup.*

Logical Form: $\lambda xyz.$woman x, walk x, near x y, table y, hold x z, yellow z, cup z

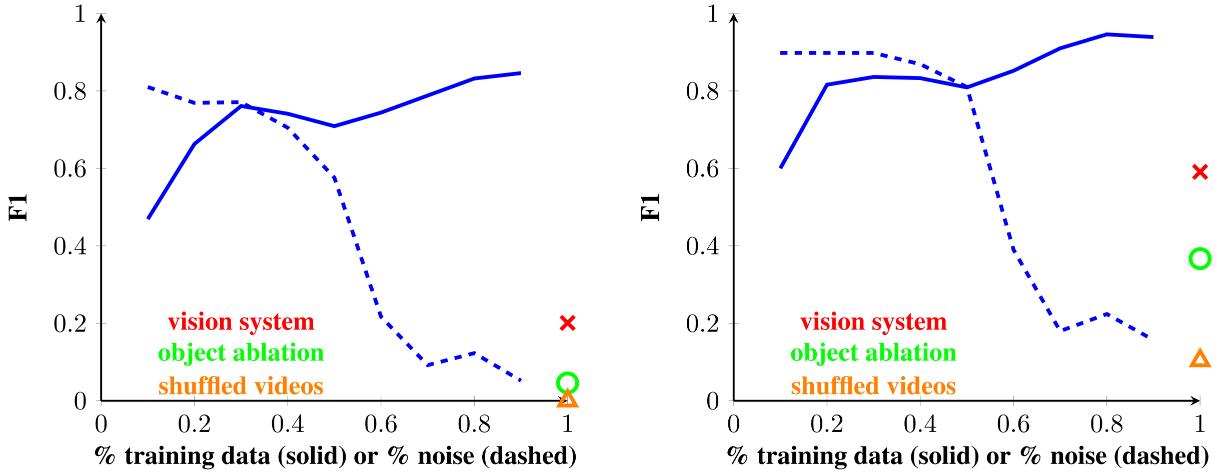
Figure 2-1: An example from our dataset of video-caption pairs. We recorded real-world videos of agents carrying out various actions while interacting with objects and used Amazon Mechanical Turk to generate captions describing the events. Using these captioned videos, we train a semantic parser on video-sentence pairs, *without access to ground-truth parses*.

$$
\begin{array}{cccc}
\text{She} & \text{takes} & \text{the} & \text{cup} \\
\hline
\mathit{NP} & \mathit{(S\backslash NP)/NP} & \mathit{NP/N} & \mathit{N} \\
\lambda x.\ \text{person}\ x & \lambda fgxy.\ fx, \text{take}\ xy, gy & \lambda fx.\ fx & \lambda x.\ \text{cup}\ x \\
\cline{3-4}
 & & \multicolumn{2}{c}{\mathit{NP} \quad >} \\
 & & \multicolumn{2}{c}{\lambda x.\text{cup}\ x} \\
\cline{2-4}
 & \multicolumn{3}{c}{\mathit{S\backslash NP} \quad >} \\
 & \multicolumn{3}{c}{\lambda fxy.\ fx,\ \text{take}\ xy,\ \text{cup}\ y} \\
\hline
\multicolumn{4}{c}{\mathit{S} \quad <} \\
\multicolumn{4}{c}{\lambda xy.\ \text{person}\ x,\ \text{take}\ xy,\ \text{cup}\ y}
\end{array}
$$

Figure 2-2: An example sentence parsed into a lambda-calculus expression using a CCG-based grammar. The parse is determined by the lexicon that associates tokens with syntactic and semantic types as well as the order of function applications. Here, we acquire this lexicon and a means to score derivations.

	Precision	Recall	F1
Direct Supervision			
exact match	85.1	84.0	84.6
near miss	*94.6*	*93.3*	*93.9*
Noisy supervision (60%)			
exact match	23.5	20.1	21.7
near miss	*42.3*	*36.2*	*39.0*
Shuffled labels (direct supervision)			
exact match	14.7	12.2	13.6
near miss	*38.4*	*32.1*	*34.9*
Shuffled videos (weak supervision)			
exact match	00.0	00.0	00.0
near miss	*10.6*	*10.3*	*10.4*
Object-only vision			
exact match	5.1	4.2	4.6
near miss	*38.7*	*34.9*	*36.7*
Our full vision-language system			
exact match	22.3	18.3	20.1
near miss	*66.3*	*55.3*	*59.1*

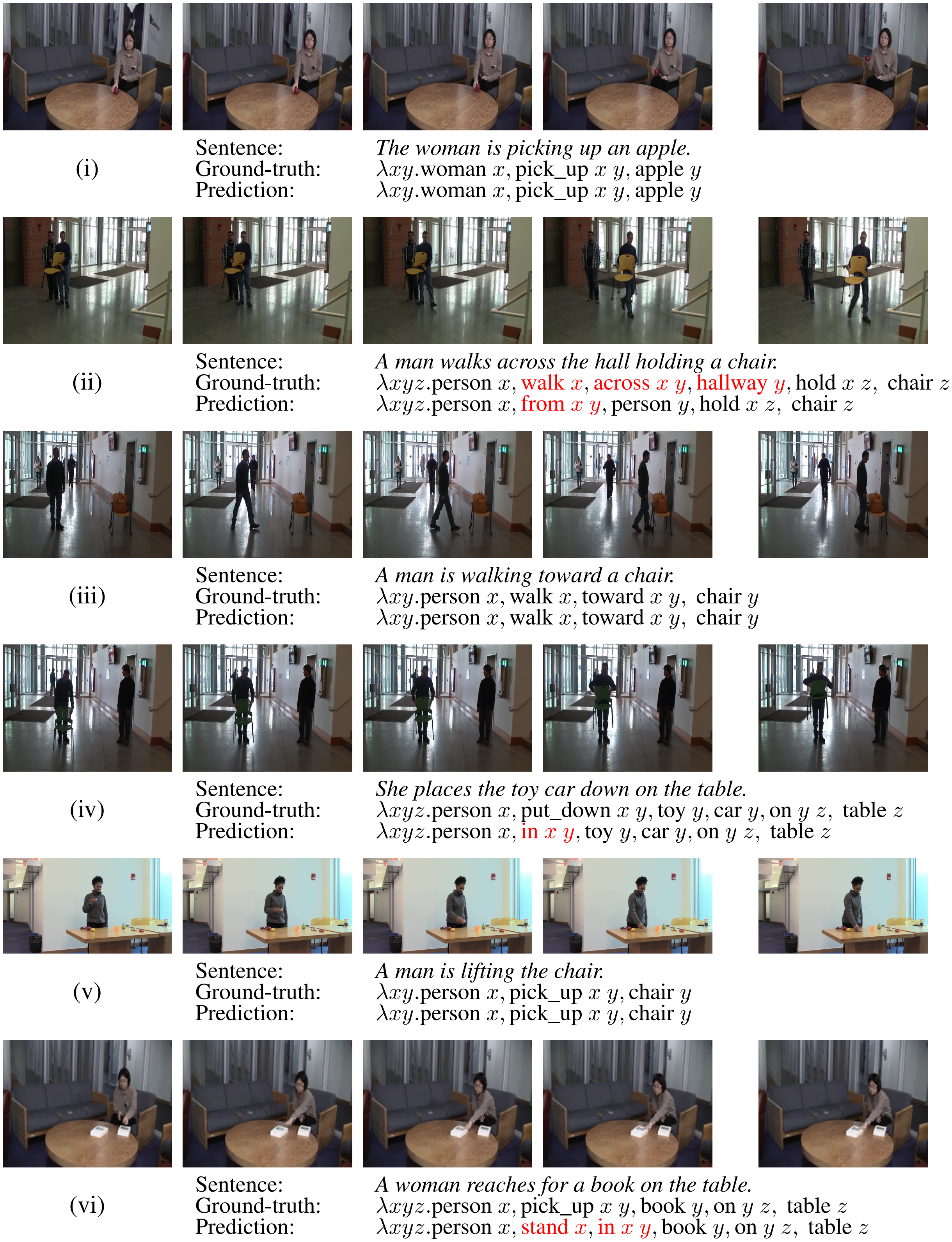
Figure 2-3: Pairs of results for each condition. On the top, we show exact match results and on the bottom, in *italics*, results for the near miss metric. In the case of *direct supervision*, we train with the target parses. In the case of *noisy supervision*, a percentage of the time (60% here) the parser randomly accepts or rejects a parse. In the case of *shuffled labels*, the target logical forms are assigned to random sentences. For *shuffled videos* the sentences are assigned to random videos. The likelihood of any sentence being true of a random video is low. In the case of *object-only vision*, the vision system consists solely of an object detector discarding any other predicates. The full *vision-language* approach learns to parse a significant fraction of sentences, far outperforming the object-only approach, and usually being within one predicate of the correct answer.



(a) **Exact match**

(b) **Near miss**

Figure 2-4: Results from training the grounded semantic parser. In blue we show , *direct supervision* as a function of the amount of training data. In dashed blue, we show *noisy supervision* that uses the whole training set but accepts and rejects parses at random for a given fraction of the time. The red cross is the full vision system while the green o is the object detector ablation. The orange triangle represents *shuffled videos* and shows chance performance. While direct supervision outperforms vision-only supervision, the grounded parser closes the gap and operates like noisy direct supervision with roughly 55% noise.



(i)
Sentence: *The woman is picking up an apple.*
Ground-truth: $\lambda xy.$woman x, pick_up $x\ y$, apple y
Prediction: $\lambda xy.$woman x, pick_up $x\ y$, apple y

(ii)
Sentence: *A man walks across the hall holding a chair.*
Ground-truth: $\lambda xyz.$person x, walk x, across $x\ y$, hallway y, hold $x\ z$, chair z
Prediction: $\lambda xyz.$person x, from $x\ y$, person y, hold $x\ z$, chair z

(iii)
Sentence: *A man is walking toward a chair.*
Ground-truth: $\lambda xy.$person x, walk x, toward $x\ y$, chair y
Prediction: $\lambda xy.$person x, walk x, toward $x\ y$, chair y

(iv)
Sentence: *She places the toy car down on the table.*
Ground-truth: $\lambda xyz.$person x, put_down $x\ y$, toy y, car y, on $y\ z$, table z
Prediction: $\lambda xyz.$person x, in $x\ y$, toy y, car y, on $y\ z$, table z

(v)
Sentence: *A man is lifting the chair.*
Ground-truth: $\lambda xy.$person x, pick_up $x\ y$, chair y
Prediction: $\lambda xy.$person x, pick_up $x\ y$, chair y

(vi)
Sentence: *A woman reaches for a book on the table.*
Ground-truth: $\lambda xyz.$person x, pick_up $x\ y$, book y, on $y\ z$, table z
Prediction: $\lambda xyz.$person x, stand x, in $x\ y$, book y, on $y\ z$, table z

Figure 2-5: Six examples of frames from videos in the dataset along with target and predicted logical forms showing both successes and failures. Failures are highlighted in red. Note how incorrect parses are usually similar to the correct semantic forms. The intended meaning is often preserved even in these cases.

Chapter 3

Learning a Semantic Parser using Temporal Logic for Robotic Planning

In this chapter, we extend our earlier work by presenting a weakly supervised semantic parser grounded in a grid world. Instead of natural language descriptions of videos, this work moves into a stateful setup where natural language commands describe actions an agent must take, conditioned on the world representation. The parser learns to map these commands to an executable formula in linear temporal logic (LTL). Similar to the CCG-based semantic parser, this parser is weakly supervised and does not have any access to the labeled annotations. We pair the semantic parser with a pretrained planner that serves as our validation function. This planner takes the generated LTL formulas and precomputed tracks of the agent of completing the command and computes the likelihood of the formula given the tracks. This work also focuses on a small dataset, at just a few thousand examples, in lieu of large-scale training.

3.1 Motivation

Children learn language by observing their environment and interacting with the world around them. In the previous chapter, we presented a semantic parser that focused on the first component- a model that receives language alongside an observation of the environment. In this chapter, we focus on the latter component- namely, interacting with and updating the world. We present a grounded semantic parser represented by a multitask encoder-decoder that receives natural language commands and environments shown as RGB images as input and produces executable formulas. The commands describe conditional, long-range temporal constraints (e.g., "*wash your hands before eating a snack; either eat grapes if they're already cut or a banana*") that are easily understood by children but cannot be represented by formalisms like are based on first-order logic. To this end, we represent formulas in linear temporal logic (Pnueli, 1977), which can represent a wide range of temporal language. The LTL formulas generated by the parser are then validated by a robotic planner, which computes a compatibility between the formula and a set of predetermined action trajectories. Notably, the parser only observes the language being used (i.e., what was said, in this case the command) and how well this language fits the trajectory of the agent (i.e., how well the formula matches observed tracks in a given environment).

Our model uses a few different sources of information to learn to map the commands to LTL formulas. First, the formulas generated by the parser are verified to be syntactically valid LTL formulas. This is a relatively common well-formedness constraint that simply ensures parses, regardless of what they hypothesize semantically, will be executable. Second, the parser has an incentive to create interpretations that are both generic enough to not overfit certain idiosyncrasies of a given environment yet also specific enough to actually capture the behavior that is observed. This is addressed through the planner, which uses three different environments and respective action trajectories to compute the likelihood of the formula. Lastly, we encourage the encoder to learn expressive encodings by using a multitask framework; we include a second decoder that acts a language generator and attempts to reconstruct the input sentence. The goal here is to maximize the knowledge conserved when translating sentences into some formalism.

For our training and evaluation data, we use a grammar to generate sentences paired with LTL formulas. We also use Mechanical Turk to get more linguistically challenging sentences that capture the same meaning but are generated by human annotators. The annotators saw a set of (disjoint) examples of commands and environments and then were given a new set of environments and asked to describe a command for the agent. The human-generated commands provide a larger vocabulary and more linguistic diversity than those from the grammar alone. Despite the human sentences being much more varied and complex, including structures which our formalism cannot exactly represent, the semantic parser still finds whatever latent structure is present and required to execute the natural-language commands produced by humans with nearly the same accuracy as those produced by machines.

Executing LTL commands and more broadly understanding temporal constraints in natural language is a challenging task. LTL works well for our domain but is just a stepping stone. It remains an important open question in grounded robotics: what representation will be enough to capture the richness of how humans use language? For instance, notions such as modal operators to reason about hypothetical futures will likely be required, but what else we may need is unclear. Having a general-purpose mechanism for learning to execute commands is extremely helpful under these conditions; we can experiment with different logics and domains with the same agents by changing the priors and planners while leaving the rest of the system intact.

Our contributions are as follows. First, we present a semantic parser that maps natural language commands to formulas in linear temporal logic without access to any annotated formulas. Next, we present a dataset of commands paired with environments that are a variant of Craft (Andreas et al., 2017); this variant is suitable for grounded semantic parsing experiments. Finally, we show a “recipe" for creating grounded semantic parsers for new domains that results in executable knowledge without annotations for those domains.

3.2 Related Work

Many approaches to training weakly supervised models use an execution-based validator for training. Berant et al. (2013) train a model on question-answer pairs using a knowledge base, where the generated formula is executed and an answer is retrieved and used as supervision instead of providing the ground-truth formula directly. Kwiatkowski et al. (2013) and Berant and Liang (2014) use a paraphrasing-style approach, where questions are first mapped to a domain-independent representation and then mapped to an executable form. The answer to this executable form is used as the supervision. For Artzi and Zettlemoyer (2013), instead of using knowledge base or question-answer pairs, their approach uses a deterministic robotic environment where commands are mapped to executable logical forms. The end state after executed the generated logical form is compared to the goal state and this end state is used as supervision. Our approach is similar, with a key difference being that our executor is non-deterministic and therefore much noisier and the input commands in our framework are highly temporal. Hermann et al. (2017) also ground their parser in an interactive world; their approach focuses on object retrieval, grounding attributes in natural language to attributes of the object. The commands are relatively straightforward, similar to Matuszek et al. (2012) but in an interactive instead of static setting. Perhaps the most similar approach to our work is that of Patel et al. (2019), who also map natural language commands to LTL formulas. A key difference here is that their work requires domain-specific LTL knowledge, whereas our model only rejects formulas that are not well-formed. The supervision for their parser comes from executing the LTL formula and receiving feedback at each step whether the agent has failed.

3.3 Notation

Natural language sentence is represented as series of *N* tokens, written as $t_{1:N}$ or $\boldsymbol{t}$. A single token at index *i* is referred to by symbol t_i. In vector notation, the series of tokens is written as $\boldsymbol{t}$. Sequences generated by the model are written with a circumflex, for instance $\hat{\boldsymbol{t}}$. In our framework, we denote the natural language input commands as $\boldsymbol{x}$ and reconstructed commands as $\hat{\boldsymbol{x}}$. The list of *k* execution traces (which are demonstrations of the command) are denoted E and a single execution trace as e or e_i. Ground-truth LTL formulas are denoted $\boldsymbol{z}$ and predicted formulas as $\hat{\boldsymbol{z}}$. The output vocabulary Z refers the potential LTL tokens that can be predicted at the parser decoder's output layer. The action space Y refers to the space of actions the agent can take in the environment (e.g., left, up, right, etc.).

3.4 Task

Given a dataset $D = \{(\boldsymbol{x}, E)\}$ consists of English commands $\boldsymbol{x}$ paired with execution traces E, we train a semantic parser to map commands to executable LTL formulas. The executions E are comprised of k trace-environment pairs: $E = \{\boldsymbol{y_i}, e_i\}_{i=1}^{k}$. A trace is a sequence of actions e.g., $\boldsymbol{y_i} = [left, right, grab, up]$ that the agent can take that satisfy the constraints of the commands and environment. The goal of the model is map natural language commands to LTL formulas that, when executed by the planner described in Section 3.6.2, satisfy the constraints of within the command. As an additional task, the model must also reconstruct the natural language command from the compressed encoded representation.

3.5 Dataset

We built a dataset of 1000 examples consisting of 100 natural language commands and 3000 environment traces to train and evaluate our parser. We generate commands and formulas from a grammar and commands from paid human annotators. Each example is defined as tuple $(\boldsymbol{x}, E)$, where $\boldsymbol{x}$ is the natural language command (either from the grammar or from annotators) and *E* is list of *k* environments.

3.5.1 Grammar-Generated Commands

First, for the grammar-generated commands, we sample sentences and formulas from a synchronous context-free grammar by following an approach similar to Jia and Liang (2016) and Goldman et al. (2018). We use the six hierarchical partially overlapping properties of LTL developed by Pnueli and Manna (1990) for diversity across LTL formulas. This sampling process across the properties is important as most randomly generated LTL formulas are uninteresting and do not express complex semantics. This parallels how most instances of the Boolean satisfiability problem SAT are uninteresting as well (Horie and Watanabe, 1997). These six categories are *safety, guarantee, persistence, recurrence, obligation, and reactivity*. Safety ensures that a property will always hold; guarantee ensures that a property will hold at least once; persistence ensures that a property will always hold after a certain point; and recurrence ensure a property will hold repeatedly over time. Obligation and reactivity are compound classes formed by unrestricted Boolean combinations of the safety and recurrence classes respectively. In our framework, we do not include negation. We further detail the components of our logic in Figure 3-1.

	Constituents	*Description*
Logical operators	$\wedge, \vee$	and, or
Temporal operators	$\Box, \Diamond, \cup$	eventually, always, until
Objects	`APPLE, ORANGE, PEAR`	objects that can be held
Relations	`CLOSER APPLE, CLOSER ORANGE, CLOSER PEAR`	spatial relations
Destinations	`FLAG, HOUSE, TREE`	destinations

Figure 3-1: We show the logical constituents for the grammar and their generic descriptions. We renamed predicates from the Craft domain for more human-readable labels. We also exclude negation.

Next, to get the execution traces, each LTL formula is first converted to a Büchi automaton using Spot (Duret-Lutz et al., 2016). Then, for each of the 1000 sampled formulas, we sample three different environments represented as 7x7 grids. Our commands and environments are given in the context of the Minecraft-like *CRAFT* adapted from Andreas et al. (2017). These grids are randomly populated with objects and landmarks, of which some are relevant for the commands and others are just distractors. The distractor items are included in the environment with probability $p = 0.5$. We use an oracle to search the action space and generate action sequences until we find a sequence that is accepted by the automaton representing the LTL formula. In this work, we only consider LTL formulas for our dataset that accept finite action sequences for horizon $t = 15$. This leads to three execution pairs, where an execution pair defined as the environment and action sequence, per natural language command, giving 3000 sentence/execution pairs total.

This process of sampling action sequences until they satisfy the command means the trajectories in our data, unlike a lot of prior work, are not necessarily the gold trajectory. Our paths are guaranteed to accept but are not guaranteed to be the most efficient, which adds an additional interesting component to the validation function. Another aspect of note is that during sampling, some commands were immediately violated by the start state for the given environment. For example, the command "always hold the gem" can only be satisfied if the initial sampled environment has agent holding the gem. A sampled environment that satisfies these initial constraints may never be found. To address this issue, we replace each predicate p with $\texttt{closer}(p) \mathsf{U} p$, where $\texttt{closer}(p)$ means "closer to p". This essentially just softens the constraint that p immediately hold, instead constraining the agent to move closer to p until p is satisfied. This does not change the semantics of the command and instead allows the agent to use the first few actions to reach a satisfying state. We also ensure the commands cannot be trivially solved and are not true from the initial state. For instance, a command like *Eventually go to the house* would be solved from the initial state if the agent were already at the house. We reject these examples.

3.5.2 Human-Generated Commands

While using the grammar is helpful efficiently generating commands and for creating LTL formulas, we are also interested in more sentences from humans that are less systematic and closer to what we expect children to hear. The way a human annotator might describe a command will likely differ from one built from a grammar, both in efficiency and in how systematic the sentences appear. We therefore took the three execution traces corresponding to each command and asked annotators to write a command describing the agent's actions across all executions. To demonstrate command-centered language (e.g., *"Go to the tree and pick up the pear"* instead of *"an agent is going to the tree and then it picks up the pear"*), we showed a disjoint batch of examples to the annotators at the beginning of the task. We collected another 1000 sentences from these annotators. Examples of environments and commands are shown in Table 3.1 and statistics of the dataset are shown in Table 3.2.

We note this process did at least in part mildly skew the annotators' vocabulary and likely explains why some annotators using words like "ensure" that are LTL predicates and may be less common in everyday language. Overall, the sentences from human annotators were both shorter on average and comprised a larger lexicon. The shorter command length could be because humans find more efficient means of communication, in line with the Rational Speech Act (Frank and Goodman, 2012) model. There are also cases where the human command does not fully capture the intended meaning from the executions. Looking at the example on the bottom row of Table 3.1, the annotator didn't grasp the concept of being near the tree before going to the house. The difficulty with fully capturing all of the constraints from just three sampled environments and execution traces is an additional point to note, meaning that our grammar-generated and human-generated commands will sometimes slightly differ in meaning from the same execution sequences.

3.6 Model

3.6.1 Multitask Encoder-Decoder

Our model is an LSTM-based encoder-decoder trained in a multi-task manner over two objective. We use a single encoder that maps the natural language command $\boldsymbol{x}$ to the hidden representation and two decoders, the *parser* and the *generator*, that each take the hidden representation as input. The parser is responsible for mapping the hidden representation to the target LTL parse $\hat{\boldsymbol{z}}$, while the generator is responsible for mapping the hidden representation back to the original natural language command $\hat{\boldsymbol{x}}$, essentially acting as an auto-encoder. A diagram of the model is shown in Figure 3-2.

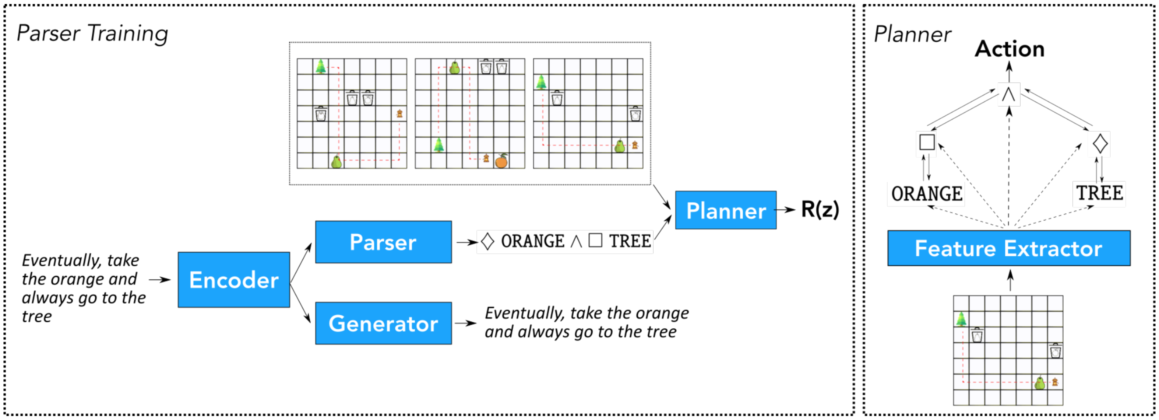
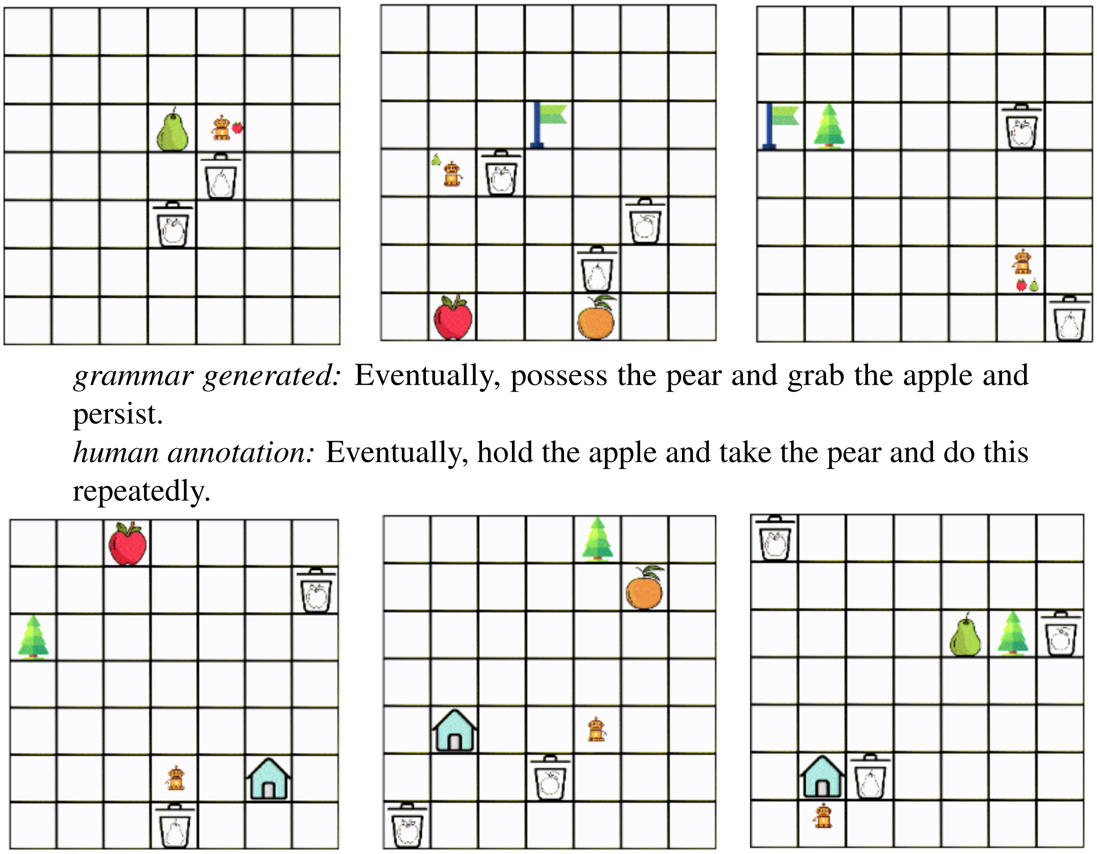


Figure 3-2: An example of the model described above. On the left, the encoder-decoder architecture is shown where the model takes a natural language command x as input and produces both executable LTL formulas (output of the parser) $\hat{z}$ and the reconstructed natural language command (output of the generator) $\hat{x}$. On the right is a depiction of the planner from Kuo et al. (2020a) and described in more detail in Section 3.6.2.



grammar generated: Eventually, possess the pear and grab the apple and persist.
human annotation: Eventually, hold the apple and take the pear and do this repeatedly.

grammar generated: At some point, start to be around the house or go to the tree and keep doing it.
human annotation: Ensure that you visit the house.

Table 3.1: Two commands (top row and bottom row) alongside three sampled environments each. For each command, we show the original grammar-generated sentence and the human annotation from six human annotators. The human-generated command is much shorter than the one from the grammar; the human generated commands were shorter on average. This could be in part due to the efficiency of human communication or due to a misunderstanding of the underlying semantics of some commands, whereby components necessary for the execution were left out.

Total # sentences	2000
# Grammar-generated sentences	1000
# Guarantee	204
# Safety	264
# Recurrence	243
# Persistence	214
# Obligation	52
# Reactivity	23
Avg. words/sent.	17.7 ± 8.4
# Lexicon size	44
# Human sentences	1000
Avg. words/formula	5.2 ± 2.9
Avg. words/sent.	8.3 ± 3.3
# Tokens in lexicon	266

Table 3.2: General dataset statistics of both the grammar and human generated commands. Note that the human-generated data is far more varied with a much larger lexicon.

Encoder

The encoder is a bidirectional LSTM with word embeddings initialized from pretrained English GloVE embeddings (Pennington et al., 2014). The encoder maps the input sequence of *N* tokens $x_{1:N}$ to the hidden representation $h_{1:N}$, that can be viewed as a high-dimensional feature representation of the input. Following the sequence-to-sequence framework, these hidden states are passed to the two decoders.

Parser Decoder

The parser decoder, herein the decoder, takes hidden representation $\boldsymbol{h}$ and generates a hypothesized LTL formula $\hat{\boldsymbol{z}}$, where each token $\hat{z}_i \in Z$. Given context c_i, hidden representation h_i and model weights W, the probability distribution for the next token is computed as:

$$p(\hat{z_i} \mid \widehat{z_{<i}}, \mathbf{x}; \theta_{\text{parse}}) = \text{softmax}(W[\widehat{z_{i-1}}; c_i; h_i]) \tag{3.1}$$

Following the approach of Guu et al. (2017), we use a stack that keeps track of the generated tokens in postfix order based on the fixed arity of LTL operators. We use this stack to constrain the valid next tokens, ensuring the generated LTL formulas are well-formed. No other specific properties of LTL or knowledge about the semantics of language are included in this constraint; we are only ensuring syntactic correctedness at the generation step. An alternative approach would be to instead reject syntactically invalid parses after generating the entire sequence. We also use epsilon dithering to encourage greater exploration during generation. Practically this means that at each time step t_i, with probability $p = \epsilon$, we follow the normal sampling procedure of selecting the next token z_{i+1} according to the softmax over output tokens. With probability $p = 1 - \epsilon$, the next token z_{i+1} is instead just sampled uniformly over all valid output tokens in Z, where the valid tokens are again only constrained to ensure well-formedness.

Parser Generator

The parser generator, herein the generator, decoder generates $\hat{\boldsymbol{x}}$, an attempt to reconstruct the natural language command $\boldsymbol{x}$ from the hidden representation $\boldsymbol{h}$. This is often used in unsupervised neural machine translation (NMT) (e.g., in Artetxe et al. (2018)) and semantic parsing can be framed as a form of NMT that maps between natural language and a formalism instead of between two natural languages.

3.6.2 Planner

We adopt the planner introduced in Kuo et al. (2020b). In its original implementation, the planner takes an LTL formula $\boldsymbol{z}$ and an environment *e* and produces action trajectories that form a satisfying path of the agent. The planner extracts features from the environment (represented as RGB image) in 5x5 patches around the agent using a CNN. These features alongside the LTL formula are mapped to a hierarchical RNN and the output of the RNN is used to predict the next most likely action. Given a potential execution trace $\boldsymbol{y}$ that is a series of actions that satisfying the LTL formula z, the likelihood of the generated formula can be computed according to $\hat{\boldsymbol{z}}$, $\mathcal{L}(\boldsymbol{y}|\boldsymbol{z}, e)$. For our model, we focus on this, namely computing a likelihood of our generated LTL formula $\hat{\boldsymbol{z}}$ given the set of *k* execution traces. This likelihood $\mathcal{L}$ is likely to be noisier and weaker signal than if we were to use guaranteed optimal paths, as our execution traces are just sampled until acceptance.

3.7 Training

Our model receives a natural language command $\boldsymbol{x}$ and execution traces E as input and generates both an LTL parse $\hat{\boldsymbol{z}}$ and a reconstruction of the natural language input $\hat{\boldsymbol{x}}$. We refer to the learned parameters of the encoder, parser and generator as θ_{enc}, θ_{parser} and θ_{gen}, respectively. For the objective function for the generator, we use a reconstruction loss computed according to

$$\mathcal{L}(z, \hat{z}) = -\sum_{x} log\ p(\hat{x}|x, \theta_{gen}) \tag{3.2}$$

For the objective function for the parser, we compare two different training schemes, REINFORCE (Mnih et al., 2016) and iterative maximum likelihood. In both setups, we use a reward based on the score from the planner. We jointly optimize both the generator and decoder losses. For the encoder and decoder, we set the hidden dimension to 1000 and use a dropout probability of $p = 0.2$. We train using the Adam optimizer (Kingma and Ba, 2015) with a learning rate $lr = 1e^{-3}$ and a batch size of 128. We set epsilon $\epsilon = 0.15$ and use $k = 3$ for the number of execution traces. During beam decoding at inference, we set the beam width to 10.

3.7.1 REINFORCE

For the REINFORCE algorithm (Williams, 1992), we learn policy parameters and treat the policy as the distribution over the output tokens. Here, since we are referencing the parser, the output tokens are the LTL symbols. The expected reward for the policy gradient is computed according to:

$$J_{RL} = \sum_{(\mathbf{x},E)\in D} \sum_{\hat{\mathbf{z}}} R(\hat{\mathbf{z}}, E) p_{\theta} \left(\hat{\mathbf{z}} \mid \mathbf{x}\right) \tag{3.3}$$

We use Monte Carlo sampling to learn the parameters. We also use epsilon dithering by randomly injecting noise into the distribution to encourage exploration.

3.7.2 Iterative Maximum Likelihood (IML)

In the second approach of iterative maximum likelihood (IML), we directly learn the parameters θ_{enc} and θ_{parser} following Liang et al. (2017) and Agarwal et al. (2019). IML essentially works in two steps: first a sampling step where we find pseudo-gold examples based on the high scoring generations and second, a gradient update step where treat the generations as gold as use maximum likelihood. For the sampling step, we generate *K* candidate formulas generated for input sentence $\boldsymbol{x}$ and compute the reward for each formula using the planner. We keep the highest scoring generated formula for each sentence that has a nonzero reward; e then train in a standard MLE framework. We repeat this process iteratively, sampling candidates then doing 10 gradient update steps. We show pseudocode for this process in Algorithm 1.

```
$\forall$x. Initialize $\mathbf{z}^*_x = \emptyset$ ;
for $n = 1$ to $N$ do
    for $(\boldsymbol{x}, E) \in D$ do
        $\hat{Z} \leftarrow \{\hat{\mathbf{z}}_i \sim p(\mathbf{x}; \theta) \mid i \in 1..M\}$;
        for $i = 1$ to $M$ do
            if $R(\hat{\mathbf{z}}_i, E) > R(\mathbf{z}^*_x, E)$ then
                $\mathbf{z}^*_x \leftarrow \hat{\mathbf{z}}_i$
            end
        end
    end
    $\theta \leftarrow \texttt{MLE}(\{\mathbf{x}\}, \{\mathbf{z}^*_x\})$
end
```

Algorithm 1: Pseudo-code implementation of our training using iterative maximum likelihood. Recall that $(\boldsymbol{x}, E)$ is an example from the dataset where $\boldsymbol{x}$ is a natural language command and E is a set of *k* executions. Generated LTL formulas are $\hat{\boldsymbol{z}}$ and the set of generations across the entire dataset is $\hat{Z}$. *M* refers to the number of execution steps for a given execution trace E that is used to normalize the reward *R*.

3.7.3 Reward

Recall that every formula can be represented by a Büchi automaton *a*. An example automaton for the given command *"Always move closer to the apple until you take the apple"* and corresponding LTL formula $\square(\text{CLOSER\_APPLE} \cup \text{APPLE})$ is shown in Figure 3-3. For a generated LTL formula $\boldsymbol{z}$, corresponding automaton $\hat{a}$ and k execution sequences E, the reward is computed according to

$$R(\hat{\mathbf{z}}, E) = \begin{cases} \frac{1}{k}\sum_{(e_i,\mathbf{y}_i)} \frac{1}{|\mathbf{y}_i|}\sum_j \mathcal{L}(\mathbf{y}_{i,j} \mid \hat{\mathbf{z}}, \mathbf{y}_{i,<j}, e_i) & \forall(\mathbf{y}, e) \in E.\mathbf{y} \in \hat{\mathbf{a}} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where $\mathcal{L}$ is the likelihood computed by the planner. The reward is the normalized across the k executions and the number of steps in each execution trace $e \in E$. If any of the execution sequences are rejected, the reward is zero. This helps to ensure that the generated formula is actually capturing all of the constraints as represented different executions and not just overfitting to a single execution. Because these sequences are just sampled, there is not a guarantee that all constraints are covered so this still an imperfect metric.

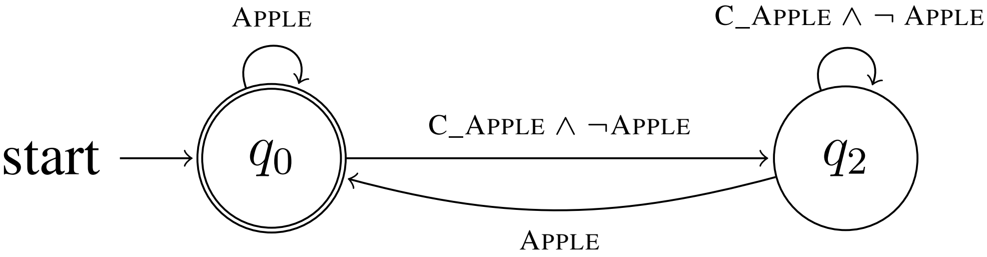


Figure 3-3: A nondeterministic finite state automaton for the LTL formula $\square$(CLOSER_APPLE $\cup$ APPLE) that corresponds to the natural language command "*Always move closer to the apple until you take the apple*". LTL formulas can be automatically converted to Büchi automaton using off-the-shelf algorithms; in this work, we use the Spot (Duret-Lutz et al., 2016). For this example, the predicate APPLE is true when the agent is holding the apple. The predicate C_APPLE is true when the robot is moving towards the apple. Any execution that violates the "always" constraint by either moving away from the apple or failing to pick it up will be rejected.

Equation (3.4) emphasizes two factors, namely permissiveness and efficiency, in computing this reward. The formula needs to balance not being overly permissive, where executions that do not actually satisfy the constraints are accepted, while also being efficient, which means not including unnecessary predicates and operators. As a representative example, take the natural language command "*Go to the house and then go to the tree*" and a generated formula that essentially just encodes the meaning *go to the house*. This generated LTL formula is efficient, by not encoding unnecessary semantics, but is also overly permissive and accepts any execution trace that goes to the house, even if it does not go the tree. The planner therefore assigns a lower likelihood to this formula because a simpler executions that do not go to the tree could have satisfied the constraints. For the CCG parser, the equivalent concept would be CCG formulas like $person(x)$ or even more simply $object(x)$ that are true of nearly every video and are therefore overly permissive. On the other hand, efficient formulas that use as few symbols as necessary for conveying the underlying meaning and do not include unnecessary components are also rewarded by the planner by receiving a higher likelihood than a formula containing constraints not shown in the traces. This similarly ties into the features implemented in Section 2.4 about the Rational Speech Act (Frank and Goodman, 2012) regarding efficient communication. These two factors, permissiveness and efficiency, provide opposite pressures and therefore work together to encourage parser to generate optimal formulas.

Lastly, to more increase the efficiency of reward computation, we keep a cache of previously computed scores for a given formula and execution. This is similar to the caching described in Section 2.4. We also reject any formula that contains predicates that referred to objects not observed in the environment. These two changes do not incorporate prior knowledge about LTL or the specific environment and instead serve solely to speed up the process of computing the reward.

3.8 Results

We present the parsing results for both the grammar-generated commands and the human-generated commands. For the grammar generated commands, we show the results under full supervision where the model has access to the gold formula, as well as the REINFORCE and IML approaches, each with and without the generator reconstructing the original sentence. We use a combination of metrics to estimate the model's performance. First, the metric **exec** shows what percentage of formulas accept all 3 execution traces; this is an overestimate of true performance because accepting an execution trace doesn't necessarily mean the model fully understood the meaning of the command.

Plan shows how often the planner is able to produce an accepting trajectory of actions given our candidate formula. This is an underestimate of performance, given that there are multiple ways to carry out commands and even two humans side-by-side may choose different action sequences that are both equally valid and efficient. This is also evidenced by the fact that the plan column's performance is lower under supervision with the formula.

The next metric **seq** computes the F1 token match between the generated and ground-truth LTL formulas. This is a relatively strict metric that even human annotators would not perform perfectly on, because the same actions can be carried out for many different reasons and many LTL formula are equivalent in context. Our last metric is **exact**, which measures the percentage of predicted formulas that are perfectly equivalent to the ground truth. This is similar to our exact match metric described in Section 2.6.2, just for the LTL formalism instead of CCG. To compute the exact match, we convert both the generated and ground-truth LTL formulas to an automaton and test whether they accept the same languages. Like the **seq** metric, this is strict and does not necessarily test whether the two formulas encode the same meaning. This is additionally apparent given that the conversion from formula to automaton is not deterministic across different algorithms, showing that different automata can encode the same formula.

For human-generated sentences, we show use the **exec** and **plan** metrics. Because we don't have ground-truth LTL formulas for these commands since they were not sampled from an algorithm, we run a different baseline/ablation by supervising with random formulas. Again similar to the CCG-based parser project, the parser is able to still learn some structures of the formalism even with random parses. This demonstrates that there is enough systematicity in both the CCG and LTL formalism that the model can glean information even from random parses totally uncorrelated with the language input. We also see that, for both grammar-generated and human-generated sentences, having the generator does provide a small performance boost. And even though the sentences differ between the grammar-generated and human-generated sets, the performance is relatively comparable. In both cases, the parser is able to produce LTL formulas that accept the execution sequences and that the planner can leverage to make accepting action sequences. The full results are shown in Table 3.3 and an example command and the model's prediction for each training scheme is shown is Table 3.4.

grammar-generated					human-generated		
	exec	*plan*	*seq*	*exact*		*exec*	*plan*
supervised with formula	94.7	36.7	**94.9**	**91**.3	*random*	16.3	17.5
IML (no generator)	81.3	32.2	22.9	8.7	IML (no generator)	80.0	28.7
RL (no generator)	82.0	**41.3**	23.9	8.7	RL (no generator)	78.7	**40.7**
IML (full)	**85.3**	34.9	14.0	2.0	IML (full)	**83.3**	31.8

Table 3.3: A comparison of the results of our model trained using iterative maximum likelihood and REINFORCE. We show the results for grammar-generated data (left) and human-generated data (right) as well as the performance with and without the generator. For the grammar-generated data, we show the baseline performance of training under full supervision (maximum likelihood estimation) (labeled *supervised with formula*). We also present an ablation that is the performance of the parser being trained on human-generated using random formulas as supervision (labeled *random*).

natural language command	either grab the apple or the pear and hold them forever
ground truth LTL formula	$\Box\Diamond\texttt{Flag} \wedge \Box\Diamond\texttt{Orange}$
REINFORCE+generator	$\Box\Diamond(\Box\Diamond\texttt{Flag} \wedge \Diamond\texttt{Orange})$
IML+generator	$\Box\Diamond(\texttt{Flag} \vee \texttt{Orange})$

Table 3.4: An example command from our dataset, alongside the ground-truth LTL formula (as generated by the grammar) and the predictions from the REINFORCE + generator and iterative maximum likelihood + generator training schemes. The predictions here demonstrate some typical mistakes we found upon manual examination of the results. These predicted formulas are difficult to differentiate from observations of the robot's behavior, which explains why erroneously generated formulas can still accept the observed traces.

We also run an ablation where we remove the planner and instead use a binary reward that only indicates whether the automaton $\hat{a}$ representing the generated LTL formula $\hat{\boldsymbol{z}}$ accepts the execution tracks E for actions y according to

$$R(\hat{\mathbf{z}}, E) = \begin{cases} 1 & \forall(\boldsymbol{y}, e) \in E . \boldsymbol{y} \in \hat{\boldsymbol{a}} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

This is similar to the approach of Patel et al. (2019), where at each step the agent receives a 0/1 reward for whether the current step violates the LTL program. However, in this ablation the model only receives a single binary reward instead of one at each time step. This approach provides far less information than our likelihood-based approach that pressures the model to produce both efficient and expressive formulas. We show the results in Table 3.5.

For human-generated and grammar-generated data, for IML, we find a performance drop of about 5-30% across the *exec* and *seq* metrics when we remove the planner and use the binary reward. For *exact* and *seq*, we observe very small performance gains of around 1%. IML relies on a continuous reward signal to rank generations and select the pseudo-gold parses for maximum likelihood training, so this overall trend of large performance drop when removing the planner is not unexpected. This highlights that, even if we were to use a different planner, the important component is a continuous reward signal takes uses more information than just an accept/reject.

For REINFORCE, we observe slightly different results. Only the *plan* metric shows a consistent performance drop when removing the planner across all data. In *exec* the binary reward occasionally outperforms, though it is inconsistent and widely variable. Given that *exec* relies only on execution traces being accepted, and that the reward from a binary signal and our planner will be the same in these instances, it is not surprising that the model is able to and incentized to find the simplest formula that accepts the execution traces. The *plan* metric instead relies on the formulas correctly encoding the underlying meaning so the planner can generate an accepting action sequence. This again highlights why these two metrics together are important for understanding behavior. We conclude that the planner provides useful information that drives the parser to produce formulas that are not so general that they fail to elicit the correct behavior.

3.9 Discussion

We present a grounded semantic parser that maps natural language commands to formulas in linear temporal logic. Even with very minimal background knowledge about the environment and essentially no prior knowledge about the LTL formalism, the model is able to learn enough about the meaning and structure of the natural language to produce executable LTL formulas for robotic execution. We train the model using weak supervision, with no access to the ground-truth LTL formula at training time. Much like our work in Chapter 2, we leverage external information – in this case, an interactive robotic executor – as supervision. Even without any direct supervision in the form of ground truth LTL formulas, the model's grammar-generated

	exec	*plan*	*seq*	*exact*
REINFORCE – binary reward	91.3	14.4	30.5	12.0
REINFORCE – full planner	82.0	41.3	22.9	8.7
Iterative Maximum Likelihood (IML) – binary reward	58.7	24.2	15.6	1.3
Iterative Maximum Likelihood (IML) – full planner	81.3	32.2	14.0	2.0

human-generated

	exec	*plan*
REINFORCE – binary reward	82.7	13.8
REINFORCE – full planner	78.7	40.7
Iterative Maximum Likelihood (IML) – binary reward	70.0	22.4
Iterative Maximum Likelihood (IML) – full planner	80.0	28.7

Table 3.5: Results of an ablation where we compare computing the reward from planner described in Kuo et al. (2020b) versus a simpler binary reward. The binary reward is computed based on whether the automaton $\hat{a}$ for the generated formula $\hat{\boldsymbol{z}}$ accepts the execution traces. We show results for both the grammar-generated commands (top) and the human-generated commands (bottom0.

performance under weak supervision is still competitive with supervised approaches as shown in Table 3.3.

Learning under weak supervision presents both benefits and challenges. Weak supervision removes the need for directly labels providing labels, paralleling children who do not need or use direct feedback. Another large benefit is that labeling sentences with logical forms can be expensive and laborious, especially when done by hand as required for human-generated data. However, the challenges are in selecting the best mode of context for the model. In our approach, we use an interactive framework where the model generates LTL formulas and the planner computes the likelihood of the formula given observed execution traces. We found that some concepts appear quite similar in the demonstrations and are difficult to differentiate, leading to a noisy feedback signal. For instance, take the words *eventually* and *always*; the execution sequences for these two commands might appear similar (eventually going to the tree is encompassed by a demonstration that is always going to the tree). This is similar to challenges faced in the parser presented in Chapter 2, where visual concepts like *pick up* and *move* looked similar in the videos.

In our framework, we chose to incorporate the generator for reconstructing the original natural language command from the hidden representation from the encoder. This multi-task training setup did improve performance, showing a 1-4% increase over training to just decode into LTL alone. We could also consider other tasks that encourage the model to learn an efficient intermediate representation. Lu et al. (2020) showed that training a vision and language Transformer in a multitask setting improved performance; while we're focused less on transferring to multiple tasks, we are in a multimodal framework and may similarly see gains in the parser if we incorporate other tasks into training.

We also present a challenging dataset for training and evaluation that includes two sets of natural language commands for each given set of execution traces E. These sets are grammar-generated sentences and more linguistically diverse sentences generated from human annotators. For both sets of commands, the model is able to parse and generate LTL formulas with success. Additionally, the process of sampling execution traces is not particularly efficient as it requires sampling environments and actions until an accepting sequence is found. This is partially why we opted to use 3 execution traces for each command, though more traces would have provided more information to the model. Even with a smaller number of execution traces, finding a formula to satisfy them all is still nontrivial with chance performance being around 16%. This is an encouraging result with respect to the trade-off between model performance and the labor required to produce parallel corpora of sentence-formula pairs.

For our approach, we include a well-formedness constraint during generation. This is the only syntactic knowledge we include in the model and it's domain-independent, only ensuring that the model has matching parentheses and the correct number of arguments for a given predicate. We include no prior knowledge of the LTL formalism; this is an improvement over our work in Chapter 2 where, while minimal, did provide some prior knowledge in the form of the small seed lexicon. Ideally, we could also remove the well-formedness constraint and the model could learn to reject these formulas on its own during generation (i.e., by effectively assigning near-zero probability in the output distribution to invalid continuations).

We could also experiment with trying different linguistic priors in the form of the pretrained word embeddings. We use GloVE embeddings but could also consider using deeper contextualized word embeddings from ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019). Kocmi and Bojar (2017) show pretrained word embeddings do outperform random initialization for some tasks, and contextualized word embeddings outperform GloVE (Liu et al., 2020), so this would be a logical approach for us to try.

Chapter 4

Learning Generalizable, Compositional Representations in a Grounded Setting

In this chapter we evaluate the role of pretraining in learning flexible disentangled representations for systematic generalization in an encoder-decoder framework. We first separately pretrain two single-stream multimodal Transformers, (one for the encoder and one for the decoder) using 5 task-agnostic pretraining objectives and then we jointly finetune the encoder-decoder on gSCAN, a grounded navigation task. We select the pretraining tasks to not be overly catered toward gSCAN, instead aiming to learn reusable representations for the grid world, linguistic and action representation spaces. During finetuning, given a natural language command and an initial world state, our approach iteratively decodes a sequence of actions of length N by predicting a sub-sequence of actions of length M where $M << N$ The model updates and re-encodes the world state after each sub-sequence until the EOS token has been reached. We find that the pretrain-then-finetune approach with iterative decoding works well on gSCAN, performing comparably to prior work on the evaluation splits that have been mostly solved and leading to solid improvements on the novel direction and length generalization splits.

4.1 Motivation

Language is inherently compositional, as components like words and phrases can be infinitely combined to form novel meanings (Chomsky, 1965). This compositionality allows humans to both generate and understand a nearly limitless number of sentences. Children as young as preschool age have demonstrated the ability to generalize novel words and functions (Barner and Snedeker, 2008; Piantadosi and Aslin, 2016). Though language models similarly learn powerful linguistic representations, they still struggle at tasks requiring systematic and compositional generalization (Glockner et al., 2018; Finegan-Dollak et al., 2018; Jha et al., 2020; Bahdanau et al., 2019b; Hupkes et al., 2020). These struggles are apparent in pure language tasks as well as multimodal approaches.

We present a framework where we separately pretrain two multimodal Transformers – one encoder and one decoder – that we then finetune on gSCAN with a novel decoding strategy that leads to improvements on systematic generalization. Taking encoder-only checkpoints like BERT (Devlin et al., 2019) or decoder-only pretrained checkpoints like GPT (Radford et al., 2018) and finetuning them for an encoder-decoder framework builds on the work of Rothe et al. (2020). In our case, we pretrain models that we then finetune for mapping from initial world states and natural language commands to a sequence of actions. gSCAN is a synthetic navigation benchmark that focuses on rule-based and length-based generalizations during inference. The world states are represented as grids that contain both distractor objects and objects relevant to the target command. For pretraining the two models, we use five total pretraining tasks – masked language modeling, object attribute prediction, target object prediction, nearest object to the agent prediction, and masked action prediction, which we detail in Section 4.5. The pretraining tasks were designed to encourage the model to 1) learn visually grounded linguistic representations for words in the commands 2) to learn disentangled representations for the objects and their attributes that can be compositionally combined at later stages 3) to use cross-modal attention to understand the relationship between the actions the agent takes and the words in the command and 5) to learn identify the target object during the encoding stage to guide planning the action sequence during the decoding stage.

For the decoding strategy, we balance the runtime benefits of decoding the entire target sequence in one forward pass with the benefits of having more world state information by predicting a single action at each timestep and recursively re-encoding the world state. We train the model to predict sub-sequences of actions of length M, re-encoding the world state after each sub-sequence, instead of predicting the entire sequence of length N in one step. This approach is much more efficient than predicting one action at each timestep, which would require N total forward passes, and performs better than decoding all actions in a single pass on certain splits of gSCAN. Our contributions are a multimodal encoder-decoder Transformer framework that reaches state of the art performance on a number of splits for gSCAN. We show how multimodal pretraining leads to learning reusable visual and linguistic representations and to better systematic generalization.

4.2 Related Work

A number of benchmarks have been introduced to test generalization in language models. Lake and Baroni (2018) introduce the SCAN benchmark, where models must learn to map natural language commands to a sequence of actions. SCAN tests how well sequence-to-sequence models perform when the test set requires a knowledge of systematic rules (e.g., being able to understand the meaning of *run* and *slowly* when training only had examples combining *walk* and *slowly*), and their results show that sequence-to-sequence models struggle to generalize on SCAN. Kim and Linzen (2020) similarly probe the systematic generalization abilities of language models, instead testing models ability to map natural language sentences to logical forms through their dataset COGS. They show that both LSTMs and Transformers perform near-perfect when test data is in-domain with the training yet struggle significantly on out-of-domain data requiring steeper generalizations.

There are also numerous benchmarks in the multimodal space. CLOSURE (Bahdanau et al., 2019a), which builds on CLEVR (Johnson et al., 2017), is a visual question answering (VQA) dataset of visual scenes containing objects with questions designed to minimize dataset bias. In CLOSURE, the inference examples contain the same scene structure as CLEVR but use novel linguistic constructions for the questions. CURI is another VQA benchmark where the questions at test time are out-of-domain and must be answered under uncertainty Vedantam et al. (2021). Ruis et al. (2020) build on the generalization data of SCAN, presenting grounded SCAN (gSCAN) that tests systematic generalization in grounded command-following domain and is the dataset we use for training and inference in this chapter. Heinze-Deml and Bouchacourt (2020) present a LSTM-based sequence-to-sequence approach to gSCAN that encourages the agent to “think before acting", by first identifying the target object before navigating to it. We use the same approach for one of encoder pretraining tasks as well.Kuo et al. (2020a) take a structured approach of using constituency, dependency and semantic parsers to produce formal representations of the natural language command that they then use to build compositional networks based on the parse as a graph. This approach requires access to pretrained parsers, limiting its ability to generalize to ungrammatical commands or commands that are out of domain. Gao et al. (2020) propose a language-conditioned representation for the objects in the grid using message passing; in our framework, the representations of commands are always conditioned on the world state as well. Ruis and Lake (2022) use data augmentation to modify the training a set and a modular network to train on gSCAN; in their approach, they are able to show noticable improvements, though only on two of the splits.

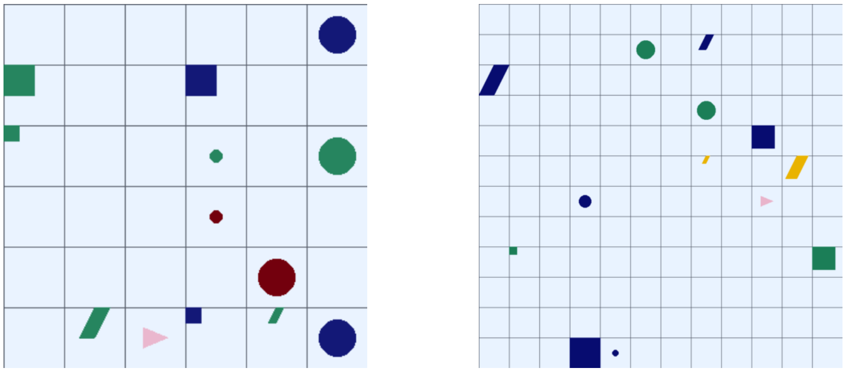
Qiu et al. (2021) present the first Transformer-based approach to gSCAN; of the prior work approaching this problem, our model architectures are the most similar. They use two-stream encoder, where the input command and world state are each encoded by separate Transformers, then a single-stream decoder attends to the encoder outputs and the action sequence. Instead of training from scratch on gSCAN, we use the pretrain-then-finetune paradigm, using pretraining objectives designed to learn disentangled representations for the language, actions and grid world. We also use a different decoding strategy, instead of the traditional approach of predicting the entire sequence conditioned on the encoder output in a single forward pass. Setzler et al. (2022) present an RNN-based approach that uses recursive decoding, where at every timestep the model re-encodes the current world state before decoding the next action. They also use random starting orientations for the agent (instead of always facing east), whcih leads to large improvements on the novel direction task; we implement this in our approach as well. We also train the model to recursively re-encode the world state, but do not re-encode at every timestep as this is time-intensive. We instead predict sub-sequences of actions of length M for a full target sequence of length N, where $M << N$. This is a different approach to decoding that is more efficient and similar to the sequence-to-sequence approach Ruiz et al. (2021), who decode in sub-steps as well.

4.3 Notation and Data

Each gSCAN example is defined as $(x, \{a_i, E_i\})$, where x is a natural language command and $\{a_i\}$ is the set of actions paired with grid world states $\{E_i\}$. The action space A is the set of actions an agent can take can complete the command, where $A = \{$*turn left, turn right, walk, stay, push, pull*$\}$. Each grid world state E_i can be represented as $d \times d \times c$ matrix, where d is the height/width of the grid and c is the total number of objects and attributes. There are two training data splits, the *compositional split* and the *target length split*. We provide further details on the type of generalization for each inference split in Appendix A. For the compositional split, the grid world is 6×6 and for the target length split, the grid world is 12×12. Objects in the grid world are defined by three attributes in their *shape* (square, circle, cylinder), *color* (red, yellow, blue, green) and *size* (1,2,3,4). Each object additionally has the latent of attribute of *weight*, which is either heavy (requiring two push/pull actions) or light (requiring one push/pull action) based on their size. Examples from both training splits are shown in Figure 4-1. We also detail each inference split in Appendix A.

4.4 Model

We train a multimodal Transformer-based encoder-decoder that takes a natural language input command, e.g., *Go to the red square*, and a corresponding grid world, and generates a sequence of actions. We describe the architecture for the encoder and decoder, the specific decoding strategy we use, and the pretraining objectives. A diagram of the full model is shown in Figure 4-2.



(a) compositional split, consisting of 6×6 grids

(b) target length split, consisting of 12×12 grids

Figure 4-1: Examples of the grid world provided as input to the model, shown here as an RGB image, for the compositional split (left) and target length split (right). The agent is represented by the pink triangle.

4.4.1 Encoder

The encoder takes a tokenized natural language command $\boldsymbol{x}$ and grid world representation E_i. The tokenized command is embedded by the *command embedding* module (see Figure 4-2) in the same manner as BERT, using word and position embeddings. We exclude token type embeddings as all tokens correspond to the same sentence. The world state E_i is a $d \times d \times c$ matrix that gets embedded by the *grid embedding module*, a multi-level CNN. A learned image token [IMG], similar to the [CLS] token for the language side, is prepended to the CNN output. The embedded command and world state are concatenated and then jointly passed to the Transformer blocks. The output of the encoder, h_{enc}, is represented by N_I hidden states corresponding to the image tokens followed by N_T hidden states corresponding to the language tokens. The hidden states corresponding to the [IMG] and [CLS] tokens, representing the image and text modality respectively, are used for classification tasks during pretraining.

4.4.2 Decoder

Decoding Strategies used by Prior Work

In a conventional encoder-decoder framework, the decoder is conditioned on the encoder hidden states h_{enc} and uses the previously generated tokens to predict the next token. This decoding in gSCAN is represented as a conditional distribution $p(a_i|a_{<i}, h_{enc})$ where a_i refers to the action at timestep i. For a BERT-style approach like that of Qiu et al. (2021), the decoder uses unidirectional attention instead of bidirectional attention and generations predictions autoregressively. Ruiz et al. (2021) modify the conventional approach and instead use an *iterative* approach, where the model decodes by generating the output in sub-steps. The conditional distribution is represented as $p(a_{i:i+m}|a_{<i}, h_{enc})$, where $a_{i:i+m}$ is a predetermined sub-sequence of the full action sequence $\boldsymbol{a}$. In their approach, the sub-steps are task-specific and require being generated from the original training examples. The other approach to decoding we'll cover is the *recursive* approach of Setzler et al. (2022) where, at every time step i, the decoder predicts a single action and then the world state is updated by the encoder to produce new encoder hidden states h_{enc}^{i+1}. The conditional distribution for an action at a_i is represented as $p(a_i|a_{<i}, h_{enc}^{i})$. For an action sequence of length N, the model makes N forward passes through both the encoder and the decoder. The model performs better on certain inference tasks as it learns less entangled representations and treats each prediction as a sub-task, but the trade-off is significantly less efficient decoding at both train and inference time.

Our Decoding Strategy

Our approach is a hybrid of iterative and recursive approaches, where we first predict a sub-sequence of actions, then update and re-encode the world state using the encoder and repeat the process until the entire action sequence has been predicted. We differ from Ruiz et al. (2021) in that we do not incorporate any task-specific priors into deciding sub-sequences, instead just deciding on an arbitrary sub-sequence length M where $1 \leq M \leq N$. We refer to a sub-sequence of actions starting at timestep i as $a_i^{sub} = a_{i:i+M}$, again of length M. Concretely, at time step i, the decoder attends to the encoder hidden states from timestep i conditioned on the command and world state E_{i-1}, the previous decoder hidden states from timestep $t-1$, and finally the previously predicted actions in the current sub-sequence a_i^{sub}. The conditional distribution is represented as $p(a_j|a_{<j}^{sub}, h_{enc}^{i}, h_{dec}^{i-1})$. The model performs a total of $\frac{M}{N}$ forward passes.

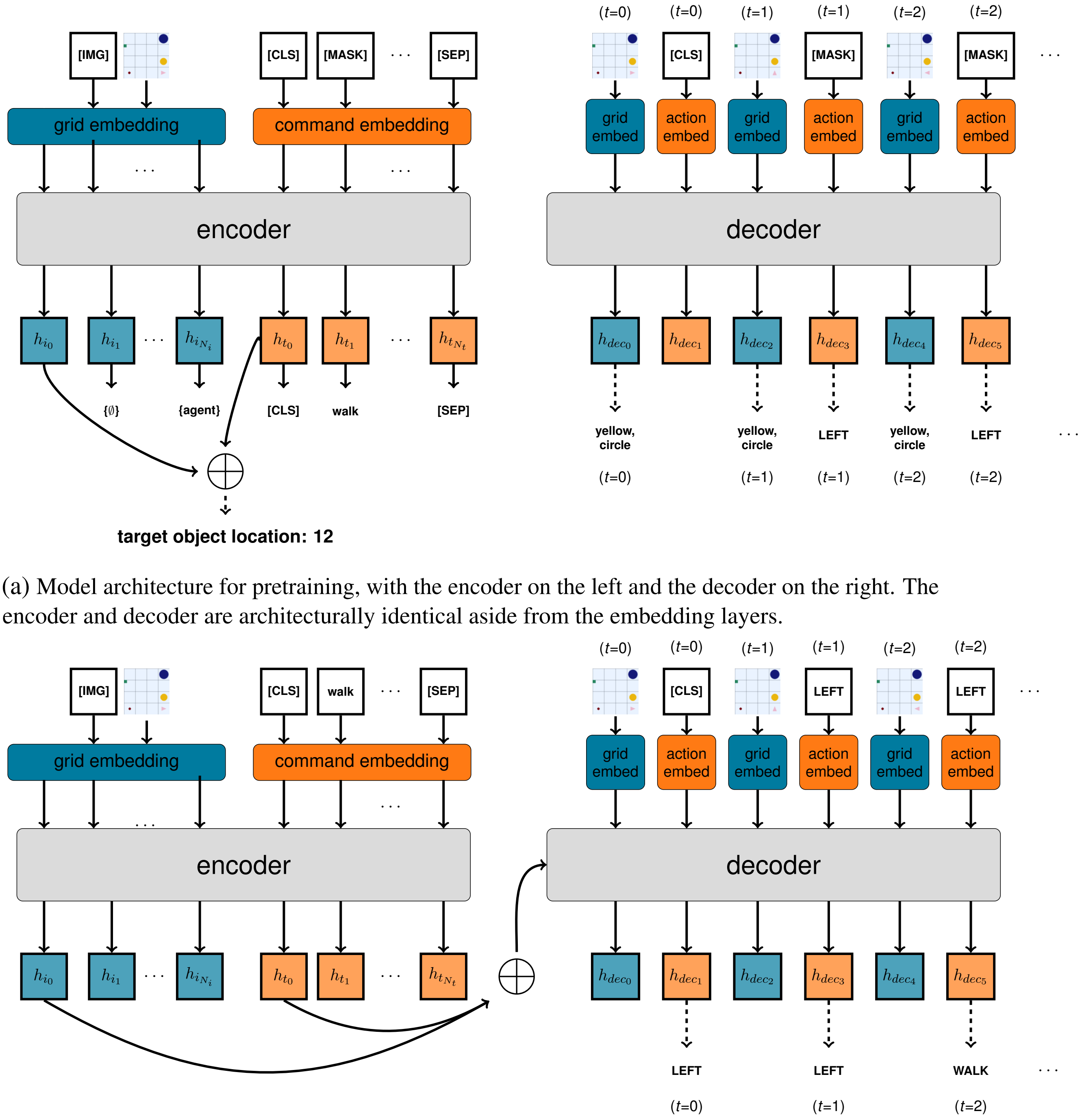
We also have one additional change that differ from prior work on gSCAN. In our approach, the model receives a compressed representation of the previous world state *at every timestep* alongside the previous actions. During training, the decoder receives ground-truth actions and world states because we use teacher forcing; during inference, the model receives the previously predicted actions and their corresponding world states. The world states are encoded in the same way as the encoder, using a multi-scale CNN. We add an additional linear layer to project from the d^2 tokens produced by the CNN (for a $d \times d$ grid) to a single token for each time step. This reduces the sequence size to make the approach more computationally feasible.

4.5 Pretraining

We pretrain the encoder using 3 pretraining objectives and the decoder using 2 different pretraining objectives. Recall that the encoder receives the tokenized input command $\boldsymbol{x}$ and the image of the initial grid E_0 as input and the decoder receives the grid world paired with the (masked) actions for each timestep. We modify the data used for encoder pretraining by randomly sampling a starting orientation for the agent; for the decoder data, we sample random starting orientation and random action sequences that do not overlap at all with the primary gSCAN training data. We have separate pretraining checkpoints for the compositional and target length splits as they have different grid sizes. The action sequence length is capped at 15 for the target length split.

Encoder Pretraining Tasks

Masked language modeling (MLM) with world state: This task is standard for vision and language Transformer models and builds off MLM introduced in BERT. A subset of the tokens in the natural language command $\boldsymbol{x}$ are randomly masked. Following the MLM approach of BERT, masked tokens are replaced with the [MASK] token 80% of the time, with another random word from the vocabulary 10% of the time, and are left unchanged the rest of the time. The model uses the bidirectional context from the other tokens in $\boldsymbol{x}$ and from the image tokens for the initial grid E_0 to predict the masked tokens. We use whole word masking, where if any masked token is a subtoken, the remaining subtokens for the


(t=0) (t=0) (t=1) (t=1) (t=2) (t=2)
[IMG]
[CLS] [MASK] … [SEP]
grid embedding
command embedding
encoder
h_{i_0} h_{i_1} … $h_{i_{N_i}}$ h_{t_0} h_{t_1} … $h_{t_{N_t}}$
{∅} {agent} [CLS] walk [SEP]
target object location: 12
[CLS] [MASK] [MASK] …
grid embed
action embed
decoder
h_{dec_0} h_{dec_1} h_{dec_2} h_{dec_3} h_{dec_4} h_{dec_5}
yellow, circle
LEFT
…
[CLS] walk … [SEP]
[CLS] LEFT LEFT …
LEFT LEFT WALK …

(a) Model architecture for pretraining, with the encoder on the left and the decoder on the right. The encoder and decoder are architecturally identical aside from the embedding layers.

(b) Model architecture for finetuning (or training from scratch) on GSCAN.

Figure 4-2: Diagram of the multimodal encoder-decoder architecture used to map from natural language commands to action sequences. While the pretraining and finetuning diagrams are shown on the same example for ease of comparison, note that the actions seen by the model during pretraining are completely random and never appear in any of the gSCAN splits.

word are also masked.

Image attribute prediction: This task focuses on learning disentangled representations for the objects in the grid. For each image token, which after embedding corresponds to a single grid cell, we classify the image representations from the final hidden layer to predict the set of attributes (color, shape and size) contained within each cell containing an object.

Target object location prediction: This task is similar to that of Heinze-Deml and Bouchacourt (2020), where the agent must identify the target object before planning the sequence of actions. We take the final hidden states corresponding to the [IMG] and [CLS] tokens from the image and text inputs and predict the location of the target object.

Decoder Pretraining Tasks

Closest object prediction: This task focuses on learning disentangled representations for the object attributes for the decoder, similar to the image attribute prediction task. Instead of predicting the set of object attributes for every cell, the model is tasked with predicting the attributes of the nearest object(s) to the agent. This also encourages the model to keep a mapping of where the agent is and what it is moving toward.

Masked action prediction: This task is similar to MLM, where the model must predict the masked actions using the surrounding context. However, instead of relying on the context of language around the masked tokens as there is not an inherent linguistic structure amongst a list of actions, we sample random actions given a starting environment, mask every action and train the model to use the compressed grid representations to understand the actions that took place.

4.6 Finetuning

The task-specific finetuning takes the command and pairs of actions and grids and learns to map to the target sequence of actions. During training, we use teacher forcing where the model has access to the ground truth action and grid at each timestep. During inference, the model receives only the start token ([CLS]) and the initial world state as input; each subsequent action and corresponding world state are produced auto-regressively. The model receives only the initial action (represented as the start token [CLS]) and the initial grid; each predicted action a_i updates the world state for the next grid E^{i+1}, which is then passed as input to the model. The prediction continues until either the stop token [SEP] is predicted or the model reaches the maximum sequence length.

4.7 Experiments

4.7.1 Setup & Model Details

We implement all code in PyTorch in the HuggingFace Transformer's library (Wolf et al., 2020). Both the encoder and decoder are 6 Transformer layers with 8 attention heads per layer following Qiu et al. (2021). The hidden dimension is 128 and the intermediate size is 256. We use the relative position embeddings introduced by Huang et al. (2020) for all implementations, which generally perform better on long sequences than absolute position embeddings. We use a single encoder for the command and world state of the encoder, also making for a slightly smaller model totaling $\sim$2.3 parameters compared to the 3M parameters of Qiu et al. (2021). The CNN kernel sizes are 1,5,7 and 9 for the 6x6 grids and 1,7,9 and 13 for the 12x12 grids. For the compositional split, the batch sizes are 64 and 128 per device for the encoder and decoder during pretraining and 16 per device for the finetuning with sub-sequence lengths of 20. For the length split, the batch sizes are 32 and 64 per device for the encoder and decoder during pretraining and 16 per device for the finetuning with sub-sequence lengths of 3. For all pretraining runs, we use a learning rate of 1e-3 with a linear warmup for the first 10% of gradient update steps and a weight decay of 0.01. For finetuning, we use learning rate of 1e-3 for the compositional split for 10 epochs and 2e-5 for the length split for 15 epochs. We train using 8 GPUs for all runs.

4.7.2 Results

Table 4.5 shows the results for the generalization splits in averaged across 3 runs with different random seeds. For the compositional split, we trained the provided examples as well as with examples modified with random starting orientations for the agent, similar

	A: Random	*B: Yellow Squares*	*C: Red Squares*
Ruis et al. (2020)	97.69 ± 0.22	54.96 ± 39.39	23.51 ± 21.82
Andreas (2020)	87.6 ± 1.19	34.92 ± 39.30	78.77 ± 6.63
Kuo et al. (2020a)	97.32	95.35	80.16
Heinze-Deml and Bouchacourt (2020)	94.19 ± 0.71	87.31 ± 4.38	81.07 ± 10.12
Gao et al. (2020)	98.6 ± 0.95	99.08 ± 0.69	80.31 ± 24.51
Qiu et al. (2021)	99.95 ± 0.02	**99.90 ± 0.06**	**99.25 ± 0.91**
Setzler et al. (2022)	99.22 ± 0.16	82.28 ± 11.5	56.29 ± 7.42
Setzler et al. (2022) (RO)	99.22 ± 0.16	82.28 ± 11.5	56.29 ± 7.42
Ruis and Lake (2022)	96.34 ± 0.28	59.66 ± 23.76	32.09 ± 9.79
Our Model	**100.0 ± 0.0**	99.5 ± 0.8	88.2 ± 14.9
Our Model (RO)	99.9 ± 0.0	99.8 ± 0.1	98.2 ± 0.4

Table 4.1: The results on gSCAN for our model compared to previous approaches across the 3 generalization splits. “RO" refers to runs where the model saw a random starting orientation for the agent.

to Setzler et al. (2022). Splits A, B, C, E and F are already “solved" splits with prior work achieving near-perfect performance (notably Qiu et al. (2021)); we reached similar performance using our approach on these splits as well. We observe the most noticeable gains on split D, the novel direction split, which had the lowest performance by far on previous approaches. Our model each 89.3% accuracy, nearly doubling the closest results from Setzler et al. (2022), which along with Kuo et al. (2020a), were the only approaches to get any examples correct during inference. For the target lengths, our model achieves high performance of ∼94% on generalization split and perfect performance on the in-domain split (sequences $\leq$ 15). It is well above the RNN baseline presented in the original gSCAN paper and also outperforms the results of Setzler et al. (2022) by about 10%.

Ablations

To test the contribution of first pretraining the model on task-agnostic objectives and then finetuning on the task-specific, command-to-action data, we compared the best approaches for the compositional and target length splits in the previous section to models trained from scratch. For a more fair comparison since pretrained models to converge more quickly, we increased the train time to 20 epochs. For the compositional split, we do not observe a distinct difference on the splits that are effectively solved. For some splits, the model

	D: Novel Direction	*E: Relativity*	*F: Class Inference*
Ruis et al. (2020)	0.00 ± 0.00	35.02 ± 2.35	92.52 ± 6.75
Andreas (2020)	0.00 ± 0.00	33.19 ± 3.69	85.99 ± 0.85
Kuo et al. (2020a)	5.73	75.19	98.63
Heinze-Deml and Bouchacourt (2020)	-	52.8 ± 9.96	-
Gao et al. (2020)	0.16 ± 0.12	87.32 ± 27.38	99.33 ± 0.46
Qiu et al. (2021)	0.00 ± 0.00	99.92 ± 1.16	99.98 ± 0.01
Setzler et al. (2022)	3.11 ± 0.87	57.99 ± 7.21	98.51 ± 0.28
Setzler et al. (2022) (RO)	43.60 ± 6.05	53.89 ± 5.39	95.74 ± 0.75
Ruis and Lake (2022)	0.0 ± 0.0	49.34 ± 11.6	94.16 ± 1.25
Our Model	0.0 ± 0.0	**99.9** ± **0.1**	100 ± 0.0
Our Model (RO)	**89.3** ± **15.5**	99.8 ± 0.0	**99.9** ± **0.0**

Table 4.2: The results on gSCAN for our model on another 3 generalization splits. “RO" refers to runs where the model saw a random starting orientation for the agent.

without pretraining performs slightly better but these splits are within a point. We do however observe a major difference on split D, which tests commands that require going in a novel direction. The difference between the model trained from scratch versus finetuned from pretrained checkpoints is more distinct for the target length split. The performance differs by 4% for target sequence lengths up to 17 and 20% overall. Recall that during pretraining for this split the random action sequences are up to, but not longer than, 15 actions total to ensure the model is not skewed by seeing long sequences before the finetuning stage.

4.8 Discussion

We present a framework for a pretrained multimodal encoder-decoder Transformer that uses a novel decoding strategy and improves performance on two gSCAN splits. First, we separately pretrain two models using 5 total pretraining tasks, inspired by the work of Rothe et al. (2020) showing that individual checkpoints can be finetuned in an encoder-decoder framework. Finetuning from pretrained checkpoints requires fewer iterations than training from scratch, making it more computationally efficient to evaluate different hyperparameters and other variations in training. Additionally, the pretraining tasks we presented are transferable to over vision and language navigation domains as well. The pretraining led to learning cross-modal, reusable representations that ground natural language

	G: Adverb k-shot, k=1	*H: Adverb to Verb*
Ruis et al. (2020)	0.00 ± 0.00	22.7 ± 4.59
Andreas (2020)	0.00 ± 0.00	11.83 ± 0.31
Kuo et al. (2018)	11.94	21.95
Heinze-Deml and Bouchacourt (2020)	-	-
Gao et al. (2020)	-	33.6 ± 20.8
Qiu et al. (2021)	0.00 ± 0.00	22.16 ± 0.01
Setzler et al. (2022)	0.00 ± 0.00	21.94 ± 0.15
Setzler et al. (2022) (RO)	0.00 ± 0.00	21.95 ± 0.03
Ruis and Lake (2022)	**80.04 ± 6.06 (k=5)**	**76.84 ± 26.94**
Our Model	0.0 ± 0.0	22.2 ± 0.0
Our Model (RO)	0.0 ± 0.0	22.4 ± 0.3

Table 4.3: The results on gSCAN for our model on the remaining two generalization splits. “RO" refers to runs where the model saw a random starting orientation for the agent. Note that the Ruis and Lake (2022)'s k-shot results for split G for are reported for *k=5*, instead of *k=1*.

	I: Length			
	$l \leq 15$	$l \leq 16$	$l \leq 17$	*overall*
Ruis et al. (2020)	94.98 ± 0.1	19.32 ± 0.02	1.71 ± 0.38	2.10 ± 0.05
Setzler et al. (2022)	-	-	-	84.42 ± 3.24
Our Model	**100.0 ± 0.0**	**98.2 ± 0.5**	**98.2 ± 0.4**	**94.2 ± 0.7**

Table 4.4: The results on gSCAN for Ruis *et al.* and our model on the length generalization split. The remaining models did not train or evaluate on this split and are therefore excluded from this table.

in the grid world and map actions to changes in world states. We also use a different approach to decoding than prior work, by generating subsequences of actions of length M and then re-encoding the world state until we reach the full action sequence. This balances the efficiency of generating entire sequences during training in one forward pass with the benefits of more world knowledge through re-encoding the world state while generating the sequence.

We find our model performs particularly well on the novel direction split and the length generalization split. For two of the splits involving generalizing an adverb to a novel verb and perform k-shot learning, the performance is still low like most prior work except that of Ruis and Lake (2022). In future work, we want to explore freezing portions of the model and only finetuning lower layers. We also want to explore different model sizes and perhaps using data augmentation to increase the size of the pretraining dataset. Using a random *compositional split using random orientation*

	A	B	C	D
w/o pretraining		**99.6 ± 0.5**	99.6 ± 0	5.1 ± 3
with pretraining	98.1 ± 0.4	98.2 ± 0.5	**99.8 ± 0.1**	**89.3 ± 15.5**
	E	F	G	H
w/o pretraining	**1 ± 0**	**99.9 ± 0**	0 ± 0	22.4 ± 0.4
with pretraining	99.8 ± 0	**99.9 ± 0**	0 ± 0	**22.4 ± 0.3**

target length split

	$l \leq 15$	$l \leq 16$	$l \leq 17$	*overall*
w/o pretraining	99.9 ± 0.0	94.7 ± 3.1	94.7 ± 3.1	74.01 ± 6.1
with pretraining	**100.0 ± 0.0**	**98.2 ± 0.5**	**98.2 ± 0.4**	**94.2 ± 0.7**

Table 4.5: We ran an additional ablation where we trained from scratch without the pretrained checkpoints. We observe a performance drop of around 6% for lengths up to 17 and around 24% overall.

starting orientation during pretraining and finetuning already represents a modification to the underlying data that shows the model more variety and demonstrates improved performance. More specific to our model, we can investigate the role of the length of the sub-sequence; there is an inherent trade-off between training and inference time and performance and we can explore finding the optimal space. We chose sub-sequence lengths of 20 and 3 for the compositional split and length splits, respectively, due to the overall sequence lengths during training. A final limitation, as noted by Setzler et al. (2022), is that this approach is able to be implemented because we can easily generate intermediate world states based on ground-truth or predicted actions. This would be much more difficult in a framework that requires significantly more time to compute the intermediate states. Co-training a model to estimate world states based on the prediction could be interesting future work as well.

Chapter 5

Measuring Social Bias in Multimodal Transformers

In the previous chapter, we explore the role of fine-tuning and a modified decoding strategy in systematic generalization. In this final work, we investigate generalization from a different angle, notably the ability of pretrained vision and language (V&L) Transformers to see beyond demographic associations in their training data. Prior work such as the Word Embedding Association Test (WEAT) and Sentence Encoder Association Test (SEAT) and has been introduced to probe the presence of social biases in word embeddings for language models. We introduce metrics, Grounded WEAT and Grounded SEAT, for multimodal models and demonstrate that three generalizations answer different yet important questions about how language and vision interact in terms of social bias in models. Because these metrics can be used to test any pretrained model, our goal is for them to be used as a benchmark for bias mitigation in future work.

When considering models to probe, we focus on BERT-style architectures that have been adapted into the vision and language domain. These models surpass previous state-of-the-art results on V&L tasks like visual question answering and NLVR2 (Suhr et al., 2019) achieved by RNN-based approaches. Similar to language tasks, these models become state-of-the-art on multimodal tasks like visual question answering and NLVR2 (Suhr et al., 2019). Alongside presenting new metrics, we also release a dataset for evaluating multimodal bias created by augmenting standard linguistic bias benchmarks with over 10k images from MS-COCO, Conceptual Captions, and Google Image Search. Dataset construction is challenging because image captioning datasets like MS-COCO and Conceptual Captions that are used to train these vision and language Transformers are themselves skewed and tend to lack diversity. This made the task of finding enough data to probe the space of biases difficult. Using this dataset and the metrics we introduce, we show that 4 large-scale pretrained models encode social bias, struggle to incorporate new evidence, and are impacted by both the language and vision modalities when learning these biases. The presence of these biases in systems will begin to have real-world consequences as they are deployed, making carefully measuring bias and then mitigating it critical to building a fair society.

5.1 Motivation

Introduced by Greenwald et al. (1998), the Implicit Association Test (IAT) is a method in psychology to measure implicit biases in humans. The IAT uses response times on a classification task to measure the strength association between concepts (e.g., *flowers* and *insects*) and attributes (e.g., *pleasant* and *unpleasant*). The IAT, alongside metrics derived from it that test preschool and school-aged children, essentially demonstrates that humans of all ages learn both benign and more harmful biases. To similarly probe biases in language models, Caliskan et al. (2017) adapted the IAT for word embeddings by introducing the Word Embedding Association Test (WEAT). Instead of measuring response times, WEAT measures distances between words in the model's embedding space. WEAT initially probed then state-of-the-art word embedding model GloVE and found its embeddings do encode social biases and that these biases parallel those of humans. Deep contextualized word embeddings such as those from ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) have been tested using WEAT as well and encode biases similar to the GloVE embeddings. WEAT has also been extended to test the encodings of entire sentences through the Sentence Encoder Association Test (SEAT) (May et al., 2019) and to the encoding of specific words in context (Tan and Celis, 2019). Beyond just biases in language, Steed and Caliskan (2021) present the Image Encoder Association Test (iEAT) for probing biases in image representations; with a method and data similar to WEAT, iEAT analyzes deep representations from unsupervised vision models and has found that harmful social biases are learned by these models too. What all of these approaches have in common is that they build on the foundations laid by the IAT to probe the encodings from models in either the language or vision domain.

We take the next step and demonstrate how to test visually grounded embeddings, specifically embeddings from large-scale vision and language Transformers pretrained on image captioning data, by extending prior work into what we term Grounded-WEAT and Grounded-SEAT. We evaluate four multimodal BERT-based Transformer models, ViLBERT (Lu et al., 2019), VisualBERT (Li et al., 2019) , LXMert (Tan and Bansal, 2019) and VL-BERT (Su et al., 2019), which we strategically selected for their architectural and training differences. Grounded embeddings are used for many consequential tasks in natural language processing, like visual dialog (Murahari et al., 2019) and visual question answering (Hu et al., 2020). Many real-world tasks such as scanning documents and interpreting images in context employ joint embeddings as the performance gains are significant over using separate embeddings for each modality. It is therefore important to measure the biases of these grounded embeddings.

Specifically, we seek to answer three questions about embeddings obtained from multimodal models. First, **do multimodal embeddings encode social biases?** As noted, many prior works have demonstrated the presence of social biases in deep representations trained from language-only input or image-only input. We are curious about multimodal embeddings, where the two modalities interact and mutually inform one another. The data used to train these multimodal models might differ enough that we find different types and magnitudes of encoded bias. Using our metrics presented in this chapter, we find equal or larger biases for grounded embeddings compared to the language-only embeddings reported in May et al. (2019). We hypothesize that this may be because visual datasets used to train multimodal models are much smaller and much less diverse than language datasets. For our second question, we ask **can grounded evidence that counters a stereotype alleviate biases?** The advantage to having multiple modalities is that one modality can demonstrate that a learned bias is irrelevant to the particular task being carried out based on the other modality. We find that the bias is largely not impacted, i.e., direct visual evidence against a bias helps little. Lastly, we ask **to what degree are biases encoded in grounded word embeddings from language or vision?** It may be that grounded word embeddings derive all of their biases from one modality, such as language. In this case, vision would be relevant to the embeddings and downstream task, but might not impact the measured bias. We find that, in general, both modalities contribute to encoded bias, but some model architectures are more dominated by language.

To probe the models and answer the above question, we curated a dataset of images and English captions, where the captions are drawn from prior bias tests for language models. We first extracted images from COCO (Chen et al., 2015) and Conceptual Captions (Sharma et al., 2018); the images and captions in these datasets lack diversity, making finding data for most existing bias tests using these datasets alone nearly impossible. To address this, we gathered additional data from Google Image Search that depicts the targets and attributes required for all bias tests considered leading to around 10k total images.

Our contributions are as follows. We present two new metrics, Grounded-WEAT and Grounded-SEAT, that are used to answer three questions about biases in grounded embeddings. While we specifically focus on vision and language Transformers, these metrics can be used for any model that has two distinct modalities that it encodes. Alongside these metrics, we present a new vision and language bias probing dataset for testing these biases in vision-and-language models. We demonstrate that grounded word embeddings have social biases (Experiment 1), show that grounded evidence has little impact on social biases (Experiment 2), and finally show that biases come from a mixture of language and vision (Experiment 3).

5.2 Related Work

5.2.1 Vision and Language Transformers

Prior to the introduction of the Transformer, most vision and language models used features extracted from a CNN, often VGG (Simonyan and Zisserman, 2015) or ResNet (He et al., 2016), to encode the image and concatenated these features with text embeddings to pass through an RNN. Transformer-based architectures for language have been adapted to multimodal frameworks by a wide range of works including VisualBERT (Li et al., 2019), ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019), CLIP (Radford et al., 2019), Unicoder-VL (Li et al., 2020a), VinVL (Zhang et al., 2021a), Oscar (Li et al., 2020b), UNITER (Chen et al., 2020), and VILLA (Gan et al., 2020). The models differ by architecture- either *single stream* where the two modalities are concatenated and then jointly encoded by a single stack of Transformer blocks or *two-stream* where two Transformer blocks seperately encode the modalities; by attention, which is either *uni-modal*, where a given modality only attends to itself, or *cross-modal*, where each modality can attend to the other; and lastly by pretraining objectives, with the most common being masked language modeling, masked region modeling (either as a classification for a masked region or a regression over masked features) and image-text alignment where an image or caption is randomly sampled a given percentage of the time similar to the next-sentence prediction objective introduced by BERT.

5.2.2 Social Bias in Humans & Word and Image Embeddings

Psychologists have developed methods to probe for intrinsic biases in humans. One such metric is the Implicit Association Test (IAT) (Greenwald et al., 1998), which measures participants' response times using a classification task for two target concepts and two attributes. These target concepts and attributes can cover relatively benign biases, like more strongly associating pleasant words with flowers and unpleasant words with insects, as well as harmful social biases, like associating women with being weak and men with being strong. These social biases have negative implications for the most marginalized people. For instance, biases based on someone's name and therefore perceived race can impact job prospects (Bertrand and Mullainathan (2004) showed applicants perceived to be Black are much less likely to receive job interview callbacks than their white counterparts). These prejudices present themselves early and have been found in preschool and elementary school children (Cvencek et al., 2011; Baron and Banaji, 2006).

Models that train word embeddings are widespread, from simpler non-contextualized approaches like bag-of-words, skip-gram and GloVE (Mikolov et al., 2013; Pennington et al., 2014) to deeper, contextualized models (Mikolov et al., 2013; McCann et al., 2017; Devlin et al., 2019; Peters et al., 2018; Radford et al., 2018). Given their appeal as reusable meaning representations, pretrained word embeddings are often used to initialize various NLP systems. This widespread use is one reason it's important to measure and ideally mitigate the presence of harmful social biases in these embeddings.

Bolukbasi et al. (2016) probe the geometry of word2vec embeddings trained on Google News for gendered stereotypes. Using a *she-he* subspace, the authors find gendered stereotypes (e.g., an analogy to *nurse-surgeon*) and validate them using participants on Amazon Mechanical Turk. While this work is not explicitly focused on identifying human-level biases as validated by formal psychology experiments, the authors do use human participants to validate that the analogies found using the geometry of the embedding space are indeed stereotypical.

Around the same time, Caliskan et al. (2017) introduced the Word Embedding Association Test (WEAT). Instead of focusing on explicit analogies, WEAT measures implicit biases by looking at the average distances in the vector space between target concepts, e.g., gender and attributes, e.g., careers and families. May et al. (2019) generalize WEAT by introducing the Sentence Encoder Association Test (SEAT) to measure biases in sentence embeddings. Contextual relationships like that of gender pronouns and career words, e.g., *He is an engineer* versus *She is an engineer*, can be measured using SEAT. Tan and Celis (2019) adapt SEAT to probe the encoding of a word in context in the sentence; first they use SEAT to encode a sentence, then they extract the representation at the model output for a specific token of interest. For instance, assuming both *He is an engineer* and *She is an engineer* are encoded by a language model, the encoding of *he* and *she* in context can be extracted and compared using their method. They also explicitly probe for intersectional biases, such as biases based on both a racial and gendered stereotype, which psychology has shown often magnify individual effects. Zhao et al. (2019) evaluated gender bias in contextual embeddings from ELMo, demonstrating that ELMo embeddings contain a bias gendered subspace when projecting the embeddings of occupations into lower dimensions and demonstrate the effects of the bias using these embeddings in a coreference resolution system. Srinivasan and Bisk (2021) evaluate gender bias in VL-BERT, using masked language modeling to probe its specific predictions given visual information.

Approaches have also been developed to analyze deep image representations and image datasets. Wang et al. (2019) show that the popular image captioning datasets has significant gender misbalance in the training data and that this misbalance is amplified by vision models. The Image Encoder Association Test (iEAT), adapted from WEAT, probes image representations obtained from unsupervised vision models Steed and Caliskan (2021). Using images from previous IAT tests, CIFAR-100 (Krizhevsky, 2009) and Google Image Search, the iEAT probes both valence (the association between two target concepts and the general concepts of pleasant/unpleasant) and more concrete stereotypes where target concepts are directly measured against strategically collected attributes. The iEAT finds that image representations encode biases similar to both humans and word embeddings.

5.3 Notation

The model input is defined by a series of N tokens, written as $t_{1:N}$ or t_1 t_2 ...t_{N-1} t_N. Tokenized input sequences are padded by a `[CLS]` token and `[SEP]` token that indicate the beginning and end of the sequence respectively. When we mention the embedding or encoding of a word or a sentence, we take this to be the output of the last layer of the model before any pretraining prediction heads (e.g., a language model head for masked language models). To get the embedding of a word (for a non-contextualized metric) or sentence, we take the embedding corresponding to the `[CLS]` token. For sets of word embeddings, we use notation A. Subsets are indicated with subscripts, for instance A_x where $A_x \subset A$.

5.4 The Grounded WEAT/SEAT Dataset

Previous biases tests for word embeddings are made of single words or sentences describing target concepts and attributes (Caliskan et al., 2017; May et al., 2019; Tan and Celis, 2019). We augment these existing text-only tests to a grounded domain by pairing each word or sentence with a corresponding image. To collect these images, we first started by searching the captions of MS-COCO (Chen et al., 2015) and Conceptual Captions (Sharma et al., 2018) for the target and attribute words from the bias tests; we chose these two datasets as they serve in some form as the pretraining data across the four Transformers we probe. We then manually checked the returned images to ensure they matched the words we searched. Note that for tests that use names for the target concepts (e.g., comparing men and women's names to careers), we instead searched captions instead for gendered terms (e.g., *men, women, she, he*) as it's extremely unlikely that names will be present in captions. All images are selected from the validation splits.

For gender, we found perhaps unsurprisingly based on work like Wang et al. (2019) that both MS-COCO and Conceptual Captions lacked widespread diversity with respect to different labels being shown across genders. For race, we were effectively unable to collect any images by searching the captions as race is rarely mentioned explicitly. When race is mentioned, it almost exclusively refers to people who are not white; this is well supported by the idea of reporting bias, that people are less likely to state properties that they don't deem novel or unusual (Gordon and Durme, 2013). For this reason, we could not build a full dataset of images to pair with the words/sentences from using MS-COCO and Conceptual Captions alone.

To compensate for their lack of diversity, we extended the dataset by gathering the images using Google Image Search. Similar to searching captions, we searched keywords and manually verified all images to ensure they were real-world images that matched the keyword being searched, keeping the top 10. Our new dataset contains 10,228 images across all sets of targets and attributes for all of the bias tests described. We show the number of images per bias test by image source in Table 5.1.

Even with the lack of diverse and plentiful images from MS-COCO and Conceptual Captions, results on these datasets are still important for two key reasons. First, we can get an indication of where pretraining datasets are lacking: the fact that images cannot be sourced for so many tests means these datasets particularly lack representation for these identities. Pretraining data serves the important function of providing enough signal for models to learn robust, reusable representations that can be fine-tuned for subsequent tasks. Skewed, biased representations from pretraining will inherently impact downstream tasks.

Second, probing using these images ensures that biases measured on the Google Image Search images are not a property of poor out-of-domain generalization. We provide original links to all collected images and scripts to download them.

Test Name	**Number of Images**
C3: EA/AA, Pleasant/unpleasant	1648
C6: Men's/women's names (M/W), Career/Family	780
C8: Science/Arts, M/W	718
C11: M/W, Pleasant/unpleasant	1680
+C12: EA/AA, Career/Family	748
+C13: EA/AA, Science/Arts	522
+Occ: M/W, Occupation	960
+Occ: EA/AA, Occupation	928
Double Bind: M/W, Competent/incompetent	560
+Double Bind: EA/AA, Competent/incompetent	440
Double Bind: M/W, Likeable/unlikeable	480
+Double Bind: EA/AA, Likeable/unlikeable	360
Angry Black Woman (ABW) stereotype (intersectional)	760

(a) Number of images for all bias tests in the dataset collected from Google Images.

Test Name	**Number of Images**
C6: M/W, Career/Family	254
+Occ: M/W, Occupation	229

(b) Number of images for bias tests in the dataset collected from COCO.

Test Name	**Number of Images**
C6: M/W, Career/Family	203
+Occ: M/W, Occupation	171

(c) Number of images for bias tests in the dataset collected from Conceptual Captions.

Table 5.1: The number of images per bias test in our dataset. Tests prefixed by "C" are from Caliskan et al. (2017); *Angry Black Woman (ABW)* and "DB" prefixes are from May et al. (2019); tests prefixed by a plus sign "+" are from Tan and Celis (2019). Each class contains an equal number of images per target-attribute pair. The abbreviation EA/AA refers to European American/African American.The dataset sourced from Google Images is complete, shown in (a). Datasets sourced from COCO and Conceptual Captions, shown in (b) and (c) respectively, contain a subset of the tests because the lack of gender and racial diversity in these datasets makes creating balanced data for grounded bias tests impractical. We renamed previous tests from male/female to men/women to reflect gender instead of sex.

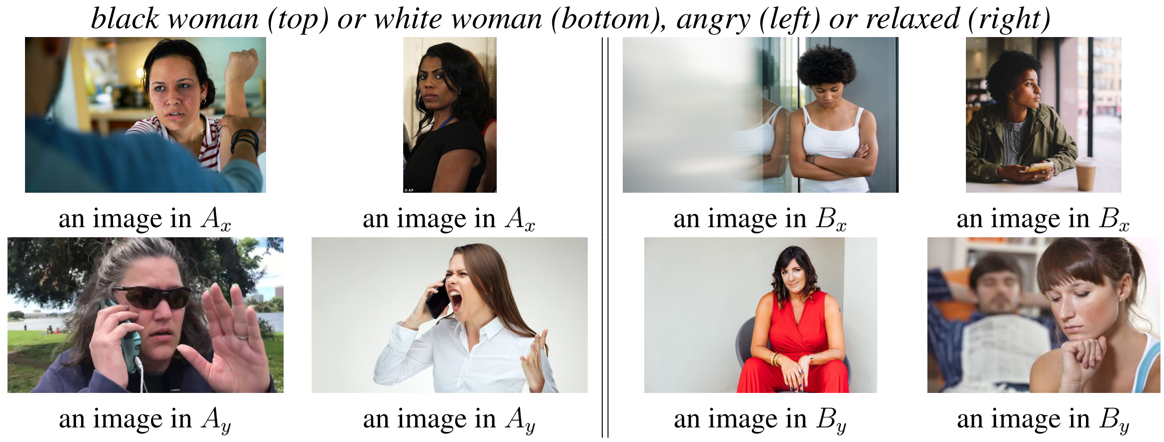


Figure 5-1: One example set of images for the bias class *Angry black women stereotype* (Collins, 2004), where the targets, X and Y, are typical names of *black women* and *white women*, and the linguistic attributes are *angry* or *relaxed*. The top row depicts black women; the bottom row depicts white women. The two left columns depict aggressive stances while the two right columns depict more passive stances. The attributes for the grounded experiment, A_x, B_x, A_y, and B_y, are images that depict a target and in the context of an attribute. The images shown here were collected from Google Image Search.

5.5 Methods

Two sets of target words or sentences, X and Y, and two sets of attribute words or sentences, A and B, are used to probe systems. For the embedding *w* of a given word, the average cosine similarity between pairs of word embeddings is used as the basis of an indicator of relative similarity, as in:

$$s(w, A, B) = \underset{a \in A}{\text{mean}} \cos(w, a) - \underset{b \in B}{\text{mean}} \cos(w, b) \tag{5.1}$$

where the function s measures how close on average the embedding w is compared to the embeddings of each attribute $a \in A$ and each attribute $b \in B$. Being systematically closer to the embeddings of A as opposed to B, or vice versa, is an indication that the concepts are more closely related in the embedding space. Such relative distances between word vectors indicate how related two concepts are and these distances are directly used in many natural language processing tasks, e.g., analogy completion (Bolukbasi et al., 2016; Drozd et al., 2016).

By incorporating both target classes X and Y and their relative distances between attributes A and B, the bias between can be measured. Measurable bias is defined as one of the two targets being significantly closer to one set of stereotypical attribute words compared to the other. The test in Equation (5.1) is computed for each set of targets, determining their relative distance to the attributes. The difference between the target distances reveals which target sets are more associated with which attribute sets:

$$s(X,Y,A,B) = \sum_{x\in X} s(x,A,B) - \sum_{y\in Y} s(y,A,B) \tag{5.2}$$

To test for statistical significance, using two equal sized sets for X and Y, the permutation test can be used to do determine the effect size, i.e., the number of standard deviations in which the peaks of the distributions of embedding distances differ, of this metric is computed as:

$$d = \frac{\operatorname*{mean}_{x\in X} s(x,A,B) - \operatorname*{mean}_{y\in Y} s(y,A,B)}{\operatorname*{std\_dev}_{w\in X\cup Y} s(w,A,B)} \tag{5.3}$$

We demonstrate how to extend these notions to a grounded setting, which requires new metrics because vision adds new degrees of freedom to what we can measure.

To explain the intuition behind why multiple grounded tests are possible, consider a trivial hypothetical dataset that measures only a single property as shown in Table 5.2. This dataset is complete: it contains the cross product of every target category, i.e., gender, and attribute category, i.e., occupation, that can happen in its minimal world. In the ungrounded setting, only 4 embeddings can be computed because the attributes are independent of the target category. In the grounded setting, by definition, the attributes are words and images that correspond to one of the target categories. This leads to 12 possible grounded embeddings[1]; see Table 5.2. We subdivide the attributes A and B into two categories, A_x and B_x, which depict the attributes with the category of target X and A_y and B_y, with the category of target Y. We shown an example of this split from our data for one of the

[1] An alternate way to construct such a dataset might have ambiguity about which of two agents a sentence is referring to, more closely mirroring how language is used. This would require images that simultaneously depict both targets, e.g., both a man and woman who are teachers. Finding such data is difficult and may be impossible in many cases, but it would also be a less realistic measure of bias. In practice, systems built on top of grounded embeddings will not be used with balanced images, and so while in a sense more elegant, this construction may completely misstate the biases one would see in the real world.

intersectional bias tests in Figure 5-1.

With these additional degrees of freedom, we can formulate many different grounded tests in the spirit of Equation (5.2). We find that three such tests, described next, have intuitive explanations and measure different but complementary aspects of bias in grounded word embeddings. These questions are relevant for both measuring bias and to understanding the quality of embeddings. For example, attempting to measure the impact of vision separately from language on grounded word embeddings can indicate if there is an over-reliance on one modality over another.

We evaluate bias tests on embeddings produced by Transformer-based vision and language models which take as input an image and a caption. These models are used to produce three kinds of embeddings (of single-word captions, of full sentence captions, and of words in the context of a full sentence sentence) that are each tested for biases. These embeddings correspond to the hidden states of the language output of each model. For single-stream models that jointly encode the vision and language input like VisualBERT and VL-BERT, these are the hidden states corresponding to the language token inputs. For two-stream models like ViLBERT and LXMERT that have a separate language and vision Transformer, these are the outputs of the language Transformer. Image features are computed in the same manner as in the original publications. When computing word and sentence embeddings, we follow May et al. (2019) and take the hidden state corresponding to the `[CLS]` token (shown in blue in Figure 5-2). When computing contextual embeddings, we follow Tan and Celis (2019) and take the embedding in the sequence corresponding to the token for the relevant contextual word, e.g., for the sentence "The *man* is there", we take the embedding for the token "man" (shown in green in Figure 5-2). Note there can be multiple contextual tokens when a contextual word is sub-word tokenized; we take the sequence corresponding to the first sub-token. When we discuss ablating modalities in Experiment 3, we use masking. To mask the language, every contextual token in the input is set to the [MASK] token. To mask the image, every region of interest or bounding box with a person label is masked. VisualBERT did not pretrain with masked regions, and is therefore not discussed in our experiment requiring region masking.

Model	*Output Sequence*
VisualBERT	`[CLS] TOK0 ... TOK_CNXT ... TOKN [SEP] [IMG] IMG0 ... IMGN`
VL-BERT	`[CLS] TOK0 ... TOK_CNXT ... TOKN [SEP] IMG0 ... IMGN [END]`
ViLBERT	`[CLS] TOK0 ... TOK_CNXT ... TOKN [SEP]`
LXMERT	`[CLS] TOK0 ... TOK_CNXT ... TOKN [SEP] [CROSS_MODAL]`

Figure 5-2: Each row shows the output sequence corresponding to a given model's output. For ViLBERT and LXMERT, we only show the output of the language Transformer. For word and sentence embeddings, we take the encoding corresponding to the `[CLS]` token; for contextual embeddings, we take the encoding corresponding to the word in context, [TOK_CNXT].

Experiment 1: Do joint embeddings encode social biases?

For our first experiment, as a proof of concept, we're measuring whether multimodal embeddings from vision and language Transformers contain social biases. This test is the most similar to WEAT and SEAT. Similarly to Equation (5.2), we compute the association between target concepts and attributes, except that we include all of the images:

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A_x \cup A_y, B_x \cup B_y) - \sum_{y \in Y} s(y, A_x \cup A_y, B_x \cup B_y)$$

To be concrete, for the trivial hypothetical dataset in Table 5.2, this corresponds to $S(1, \{5, 7\}, \{10, 12\}) - S(4, \{5, 7\}, \{10, 12\})$, which compares the bias relative to *man* and *woman* against *lawyer* or *teacher* across all target images. If no bias is present, we would expect the effect size to be zero. Our hope would be that the presence of vision at training time would help alleviate biases even if at test time any images are possible.

Experiment 2: Can grounded evidence that counters a stereotype alleviate biases?

An advantage of grounded embeddings is that we can readily show scenarios that clearly counter social stereotypes. For example, the model may have a strong prior that men are more likely to have some professions, but are the embeddings different when the visual input provided shows women in those professions? Similarly to Equation (5.3), we compute the association between the target concept and attributes, except that we include only the images that correspond to the target concept's category:

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A_x, B_x) - \sum_{y \in Y} s(y, A_y, B_y)$$

To be concrete, for the trivial hypothetical dataset in Table 5.2, this corresponds to $S(1, \{5\}, \{10\}) - S(4, \{7\}, \{12\})$, which computes the bias of *man* and *woman* against *lawyer* and *teacher* relative to only images that actually depict lawyers and teachers who are men when comparing to target *man* and lawyers and teachers who are women when comparing to target *woman*. If no bias was present, we would expect the effect size to be zero. Our hope would be that even if biases exist, clear grounded evidence to the contrary would overcome them.

Experiment 3: To what degree are biases encoded in grounded word embeddings from language or vision?

Even if biases exist, one might wonder how much of the bias comes from language and how much comes from vision? Perhaps all of the biases come from language and vision only plays a small auxiliary role, or vice versa. We make use of the same test as in Experiment 2 with the addition of masking by taking advantage of how these models are pretrained with masked language tokens and masked image regions. VisualBERT only uses masked language modeling and never masks image regions during training; it therefore cannot be probed using this method. For each test, we alternatively mask either the language tokens or the image regions that are relevant to that specific test and then measure the encoded bias. When masking image regions, we select the regions that contain people based on their labels from an object detector. As a concrete example, in test C3, we mask every name and every pleasant or unpleasant term while token masking and every person across all images while image masking. This ablates the potential bias in one modality, allowing us to probe the other.

Embedding index	Word
1	Man
2	Woman
3	Lawyer
4	Teacher

(a) Possible embeddings for an ungrounded model

Embedding index	Word	What the image shows
1	Man	*Any Man*
2	Man	*Any Woman*
3	Woman	*Any Man*
4	Woman	*Any Woman*
5	Lawyer	*Man Lawyer*
6	Lawyer	*Man Teacher*
7	Lawyer	*Woman Lawyer*
8	Lawyer	*Woman Teacher*
9	Teacher	*Man Lawyer*
10	Teacher	*Man Teacher*
11	Teacher	*Woman Lawyer*
12	Teacher	*Woman Teacher*

(b) Possible embeddings for a visually grounded model

Table 5.2: The content of a trivial hypothetical grounded dataset to demonstrate the intuition behind the three experiments. The dataset could be used to answer questions about biases in association between gender and occupation. Each entry is an embedding that can be computed with an ungrounded model, (a), and with a grounded model, (b), for this hypothetical dataset. This demonstrates the additional degrees of freedom when evaluating bias in grounded datasets. In the subsections that correspond to each of the experiments, Section 5.5, we explain which parts of this dataset are used in each experiment. Our experiments only use a subset of the possible embeddings, leaving room for new metrics that answer other questions.

5.6 Results

We evaluate each model on images from MS-COCO and Conceptual Captions for the tests where we could gather images and across all bias tests on images from Google Image Search. We compute p-values using the updated permutation test described in May et al. (2019). In each case, we evaluate the task-agnostic, pretrained base model without any task-specific fine-tuning. We leave the effect of task-specific training on biases and the role of different downstream tasks on magnifying or reducing measured biases as an interesting open question for future work.

Overall, the results are consistent with prior work on biases in both humans, language models, and unsupervised vision models. We report results for each experiment for three

Experiment 1

Gender	Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
C6: M/W, Career/Fam	W	*0.57*	*1.04*	*0.55*	*1.61*
	S	-0.18	*0.98*	*0.69*	-0.02
	C	-0.61	*0.76*	*0.17*	*0.46*
C8: Science/Arts, M/W	W	*0.77*	*0.59*	*0.43*	-0.29
	S	*0.62*	*0.26*	–	*0.19*
	C	*0.30*	-0.32	*0.13*	*0.26*
C11: M/W, Pleasant	W	-0.66	-0.91	-0.08	-1.20
	S	-0.74	-1.08	-0.20	0.01
	C	*0.42*	-0.62	*0.25*	-0.18
Competent: M/W, Competent	W	-0.23	-0.57	-1.18	-1.28
	S	-0.28	-0.29	-0.55	-1.35
	C	-0.67	0.20	-0.48	*0.31*
Likeable: M/W, Likeable	W	-1.24	-1.26	-1.10	-0.91
	S	0.10	-0.12	*0.60*	-0.03
	C	-0.42	*1.25*	-0.83	-0.19
Occupation: M/W, Occupation	W	0.02	*0.86*	*1.56*	*1*
	S	*0.77*	*0.95*	*1.32*	-0
	C	*0.98*	*1.53*	*0.52*	0.11

Race	Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
C3: EA/AA, Pleasant	W	*0.23*	*0.31*	-0.16	*1.37*
	S	*0.31*	*0.25*	*0.19*	*0.93*
	C	-0.01	-0.29	*0.44*	*0.68*
C12: EA/AA, Career/Family	W	-0.29	0.04	-0.04	-1.45
	S	-0.54	0.05	-0.32	-0.96
	C	*0.36*	*0.92*	*0.88*	0.08
C13: EA/AA, Science/Arts	W	0.04	*0.61*	*0.58*	-1.44
	S	*0.12*	*0.35*	*0.16*	*0.98*
	C	*0.58*	*1.09*	*0.92*	*0.90*
Double Bind: EA/AA, Competent	W	*0.75*	*1.28*	*0.98*	*1.44*
	S	*1*	*1.14*	*1.30*	*1.48*
	C	*1.10*	*1.19*	*1.46*	*1.54*
Double Bind: EA/AA, Likeable	W	-0.25	*0.41*	*0.93*	*0.87*
	S	-0.09	*0.73*	-0.04	*1.01*
	C	*0.97*	*1.09*	*1.40*	0.12
Occupation: EA/AA, Occupation	W	-0.15	-0.41	-0.71	1.38
	S	-0.26	-0.26	-0.40	-0.06
	C	-0.70	-0.37	-1.11	0.12
Angry Black Woman Stereotype	W	-0.07	0.41	-1.31	*1.59*
	S	-0.50	*0.46*	-0.12	-0.48
	C	*0.71*	*0.66*	*1.27*	-0.13

Table 5.3: The results for all bias classes on Experiment 1 using Google Images that asks *Do joint embeddings encode social biases?* Numbers represent effect sizes and p-values for the permutation test described in Section 5.5. They are highlighted in blue when p-values are below 0.05. Each bias type and model are tested three times against (W) word embeddings, (S) sentence embeddings, and (C) contextualized word embeddings. The answer to the question clearly appears to be yes. All models are biased. Note that out of domain, biases appear to be amplified.

Experiment 2

Gender	Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
C6: M/W, Career/Fam	W	*1.05*	*1.09*	-0.20	*1.97*
	S	-0.57	*1.34*	*0.78*	*1.57*
	C	-0.86	*0.65*	*0.21*	*0.44*
C8: Science/Arts, M/W	W	*0.77*	*0.59*	*0.43*	-0.29
	S	*0.62*	*0.26*	–	*0.19*
	C	*0.30*	-0.32	*0.13*	*0.26*
C11: M/W, Pleasant	W	-1.48	-1.33	-0.13	-0.77
	S	-1.13	-1.17	-0.55	-0.21
	C	-0.15	-0.46	*0.38*	-0.17
Competent: M/W, Competent	W	0.23	0.23	-1.37	*1.50*
	S	-0.12	-0.35	-0.98	-1.14
	C	-0.60	-0.08	-1.11	*0.44*
Likeable: M/W, Likeable	W	-1.31	-0.61	-0.93	-1.98
	S	*1.76*	-0.16	-0.81	*1.99*
	C	-0.11	*1.31*	-1	-0.12
Occupation: M/W, Occupation	W	-0.77	0.05	*1.33*	-1.74
	S	*0.33*	0.22	*0.58*	-0.20
	C	*0.90*	*1.46*	*0.34*	0.16

Race	Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
C3: EA/AA, Pleasant	W	*1.55*	*1.03*	*0.60*	*1.34*
	S	*1.54*	*0.85*	*0.84*	-0.08
	C	*0.26*	-0.14	*0.58*	*0.76*
C12: EA/AA, Career/Family	W	-0.04	*0.88*	*0.93*	-1.49
	S	*0.36*	*0.81*	*0.33*	-1.27
	C	*0.84*	*1.02*	*0.98*	*0.18*
C13: EA/AA, Science/Arts	W	-1.74	*1.27*	-0.38	-1.51
	S	-0.08	*1.04*	-0.13	*0.95*
	C	*1*	*1.39*	*0.97*	*0.96*
Double Bind: EA/AA, Competent	W	*1.13*	*1.56*	*1.06*	*1.41*
	S	*1.25*	*1.45*	*1.25*	*1.45*
	C	*1.11*	*1.20*	*1.46*	*1.57*
Double Bind: EA/AA, Likeable	W	*0.29*	*1.13*	*1.29*	*0.90*
	S	*0.42*	*1.04*	*0.43*	*1.29*
	C	*0.93*	*1.12*	*1.40*	0.06
Occupation: EA/AA, Occupation	W	-0.04	-0.48	-0.33	-1.40
	S	0.15	-0.18	0.22	-0.03
	C	-0.57	-0.19	-1.10	0.10
Angry Black Woman Stereotype	W	0.34	-0.28	-0.27	*1.67*
	S	*0.49*	-0.53	*0.31*	0.03
	C	*1.71*	*1.44*	*1.34*	-0.21

Table 5.4: The results for all bias classes on Experiment 2 using Google Images that asks *Can joint embeddings be shown grounded evidence that a bias does not apply?* Numbers represent effect sizes and p-values for the permutation test described in Section 5.5. They are highlighted in blue when p-values are below 0.05. Each bias type and model are tested three times against (W) word embeddings, (S) sentence embeddings, and (C) contextualized word embeddings. The answer to the question appears to be no, although fewer tests are statistically significant compared to Table 5.3 showing that visual evidence is helpful.

Experiment 3

Gender	Mask	VisualBERT (Google Images)	ViLBERT (Google Images)	LXMert (Google Images)	VLBERT (Google Images)
C6	T	*0.14*	*1*	*1.18*	-0
	I	–	*0.87*	*0.69*	-0.03
C8	T	*0.46*	*0.41*	*0.11*	*0.27*
	I	–	*0.39*	0.04	*0.18*
C11	T	-0.47	-1.21	-1.33	0.03
	I	–	-1.11	-0.22	0.02
Competent	T	-0.06	-0.40	-0.21	-1.99
	I	–	-0.35	-0.55	-1.05
Likeable	T	-0.07	-0.18	*0.28*	-1.99
	I	–	-0.11	*0.72*	*0.64*
Occupation	T	0.05	*1.08*	*0.92*	-0.17
	I	–	*0.91*	*1.32*	0
Race					
C3	T	*0.33*	*0.34*	*0.33*	-0.01
	I	–	*0.31*	*0.21*	*0.95*
C12	T	-0.52	0.05	-0.39	0
	I	–	0.08	-0.36	-1.06
C13	T	-0	*0.33*	-0.10	-0
	I	–	*0.33*	*0.17*	*0.95*
Competent	T	-0.44	*1.10*	*1.33*	-1.99
	I	–	*1.15*	*1.29*	*1.45*
Likeable	T	-0.68	*0.58*	0.11	-1.99
	I	–	*0.73*	-0.14	*1.06*
Occupation	T	-0.27	-0.24	-0.65	-0.17
	I	–	-0.30	-0.38	-0.25
ABW	T	*0.76*	*0.54*	-0.01	-0.42
	I	–	*0.43*	-0.13	-0.08

Table 5.5: The results for all bias classes on Experiment 3, using the second masking variant of the experiment, with Google Images asking the question *To what degree are biases encoded in grounded word embeddings from language or vision?* Numbers represent effect sizes and p-values for the permutation test described in Section 5.5. All numbers were measured over sentence-level encodings. They are highlighted in blue when p-values are below 0.05. Biases are measured for masked tokens (T) and masked image regions (I). This answer appears to be that both vision and language play a significant role, but this differs across model architectures.

		Experiment 1				Experiment 2			
Gender	Level	VisualBERT COCO	ViLBERT ConCap	LXMert COCO	VLBERT ConCap	VisualBERT COCO	ViLBERT ConCap	LXMert COCO	VLBERT ConCap
C6	W	0.13	*0.94*	*0.92*	-0.14	0.15	*0.95*	*0.61*	*1.98*
	S	*0.28*	*1.11*	*1.32*	0	*0.41*	*0.83*	*1.16*	-1.17
	C	-0.20	*0.80*	*1.53*	*0.61*	-0.99	*0.58*	*1.46*	0
Occupation	W	-0.07	*0.75*	*0.39*	-0.31	-0.64	-0.52	-0.66	*1.99*
	S	-0.23	*0.73*	-0.18	-0.01	0.09	-0.30	-1.14	*0.69*
	C	-0.32	*0.58*	-0.14	0.01	-0.35	*1.96*	-0.70	*0.90*

		Experiment 3		
	Mask	ViLBERT ConCap	LXMert COCO	VLBERT ConCap
C6	T	*1.15*	0.01	0
	I	*1.09*	*1.32*	-0
Occupation	T	*0.74*	-0.07	0
	I	*0.71*	-0.17	0

Table 5.6: The results for two classes of bias on all three experiments using COCO and Conceptual Captions. Images for other bias classes could not be found in these datasets. These results are generally consistent with results on the Google Images dataset.

types of embeddings: word embeddings, sentence embeddings, and contextualized word embeddings. While there is broad agreement between these different ways of using embeddings, they are not identical in terms of which biases are discovered. It is unclear which of these methods is more sensitive, and which finds biases that are more consequential in predicting the results of a larger system constructed from these models. Methods to mitigate biases will hopefully address all three embedding types and all of the three questions we restate below.

Experiment 1: Do joint embeddings encode social biases?

For Experiment 1, described in Section 5.5, we designed an experiment to probe whether joint embeddings from multimodal Transformer-based models encode social biases. For this experiment, the results presented in Table 5.3 and Table 5.6 show that the answer is **yes, joint embeddings encode social biases**. For every model we tested, we found multiple bias tests with statistically significant results. Numerous biases are uncovered with results

Number of statistically significant tests out of 6 total gender bias tests

	Experiment 1				Experiment 2				Experiment 3				
Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google	Mask	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
W	2	2	3	3	2	2	2	2	T	-	1	3	4
S	2	1	3	3	3	3	2	2	I	-	2	3	3
C	3	3	3	4	2	3	3	4					

Number of statistically significant tests out of 7 total race bias tests

	Experiment 1				Experiment 2				Experiment 3				
Level	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google	VisualBERT Google	ViLBERT Google	LXMert Google	VLBERT Google	Mask	ViLBERT Google	ViLBERT Google	LXMert Google	VLBERT Google
W	2	4	4	3	3	4	5	4	T	-	0	5	2
S	3	4	5	3	4	3	5	5	I	-	4	5	3
C	5	7	5	6	6	4	5	6					

Table 5.7: A summary of all previous results on the new image dataset derived from Google searches showing the number of significant bias test partitioned by the type of test. There are a total of 6 gender bias tests and 7 race bias test. Experiments 1 and 2 show no strong differences between models while in Experiment 3 ViLBERT stands out.

that are broadly compatible with May et al. (2019) and Tan and Celis (2019). It appears that more pronounced social biases exist in grounded compared to ungrounded embeddings.

Experiment 2: Can grounded evidence that counters a stereotype alleviate biases?

In humans, counterstereotypical evidence can impact implicit attitudes; we aimed to test the same hypothesis in grounded models (see Section 5.5). The results presented in Table 5.4 and Table 5.6 indicate that the answer is no. Biases are somewhat attenuated when models are shown evidence against them, but overall, **preconceptions about biases tend to overrule direct visual evidence to the contrary**. This is worrisome for the applications of such models. In particular, using such models to search or filter data in the service of creating new datasets may well introduce new biases.

Experiment 3: To what degree are encoded biases in joint embeddings from language or vision?

Recall that Experiment 3 focused on selectively masking text tokens to evaluate the contribution of language and selectively masking image regions to evaluate the contribution of vision (see Section 5.5). The results for Experiment 3 are presented in Table 5.5 and Table 5.6. We report results for the word-level encoding and sentence-level encoding, observing comparable results. We did not measure contextual embeddings for this experiment as the embeddings would include the encoding for the [MASK] token. The results indicates that biases arise from both modalities, but this does differ by model architecture. For VL-BERT language appears to dominate. For the other models, it is less conclusive if one modality contributes significantly more than the other. It could be that the biases in language are so powerful that vision does not contribute to them given that in any one example it appears unable to override the existing biases (Experiment 2).

5.7 Discussion

Visually grounded embeddings have biases similar to language-only and vision-only embeddings and adding the visual modality does not appear to help eliminate these biases. At inference time, vision has difficulty overcoming biases, even when presented counter-stereotypical evidence. This is worrisome for deployed systems that use such embeddings, as it indicates that they ignore visual evidence that a bias does not hold for a particular interaction. Overall, language and vision each contribute to encoded bias, yet the means of using vision to mitigate is not immediately clear. We enumerated the combinations of inputs possible in the grounded setting and selected three interpretable questions that we answered above. Other questions could potentially be asked using the dataset we developed, although we did not find any others that were intuitive or non-redundant.

While we discuss joint vision and language embeddings, the methods introduced here apply to any embeddings extracted from multimodal models, such as joint audio and language embeddings (Kiela and Clark, 2015; Torabi et al., 2016; Iashin and Rahtu, 2020).

Our work generally focuses on grounding in vision, but other modalities also play a role in how humans learn and encode social biases and are therefore important to test in grounded models as well. One potential drawback is that measuring bias in different grounding frameworks would require collecting a new dataset; however our metrics, Grounded-WEAT and Grounded-SEAT, can be used on any newly collected dataset to answer the same three questions.

In the current pretrain-then-finetune paradigm, language models are pretrained on large-scale dataset that are often scarcely curated, if at all Rogers (2021). This is common given how expensive training large models can be and the ease with which pretrained model checkpoints can be easily finetuned on downstream tasks. We demonstrate that going out-of-domain into a new dataset amplifies biases in our comparison of biases from in-domain pretraining data (i.e., COCO and Conceptual Captions) versus the images gathered from Google Image Search. This need not be so: out-of-domain models have worse performance that might result in fewer biases. We did not test task-specific fine-tuned models given the large number of downstream tasks and the difficulty of comparison between them, but do intend to do so in the future work. It would also be helpful to test the differences between models instantiated from pretrained checkpoints (e.g., initializing the language encoder from a BERT checkpoint) versus models that are randomly initialized.

As demonstrated by the wealth of psychology work, humans clearly have biases, not just language and vision models. Approaches to "debiasing" humans, namely being more inclusive and conscious of the way our implicit attitudes impact the way we receive and treat others, is a hugely challenging task. The same is very much true in NLP and mitigating bias in language models is a largely challenging problem. We do not address mitigation in this chapter and instead hope these metrics can serve as a benchmark in debiasing techniques.

One approach that has shown a measurable reduction of prejudice in humans is that of counterstereotypical evidence (Peck et al., 2013; Columb and Plant, 2016). Straightforward applications of this idea are far from trivial. Though not exactly counterstereotypical, Wang et al. (2019) show that merely balancing a dataset by a certain attribute is not enough to eliminate bias. Perhaps artificially manipulating visual datasets to actively include more diverse and representative data can play a role in bias reduction. We hope that these datasets and metrics will lead to understanding human biases in grounded settings as well as the development of new methods to debias representations.

One final point we would like to note is the importance of understanding gender versus sex and being thoughtful in our word choices as explore gender biases. We would like to urge subsequent work to avoid a common ethical problem we have noticed while reviewing the literature on bias in NLP. Much prior work refers about gender uses the phrases “male” and “female”, thereby conflating gender and sex. Recent work in psychology has disentangled these two concepts, and conflating them can cause harm to many people. As pointed out by Blodgett et al. (2020), research in NLP analyzing social bias in models would benefit from connecting with and grounding our work more broadly in the lived experiences of real people.

Chapter 6

Discussion

Natural language is complex and multifaceted, with the semantic meaning of words, and the syntactic with many rules and examples that go against these rules, and nuances that acquiring language remarkable. Where children are robust and essentially guaranteed to learn their native language, neural language models can be quite brittle. This thesis explored how what we know about children language acquisition -– particularly the scale and type of linguistic information children receive, the type of indirect feedback they receive, and how they generalize in light of new data – can motivate work in NLP. The contributions of this work are recapped below.

Chapter 2 presents a semantic parser trained using only captioned videos without other annotations by using a validation function based on the compatibility between the predicted parse and the video. We show how linguistic and visual ambiguities can be solved by using cues from one modality to disambiguate the other and how bootstrapping from a small number of sentence-logical form pairs can be combined with un-annotated sentences to further improve performance. We also designed and release a dataset of captioned videos, where the videos are complex with multiple agents and objects and the captions are generated from Amazon Mechanical Turk. This work highlights that using a vision system with an understanding of agents and objects and the relationship between them is powerful enough to learn a semantic parser without the need for labels or a large dataset.

An interesting source of error in the experimental results comes from visual ambiguities. In evaluating where our model failed to correctly parse sentences, we found that many predictions uses an incorrect predicate that has a similar meaning to the target predicate. Examples such as *hold the apple* and *move the apple* for instance are difficult to visually disambiguate; the bounding boxes tracking the apple will look similar in both instances and will therefore receive similar likelihoods from the Sentence Tracker. Improvements in the visual representation – either from better object detectors or using a different approach like semantic segmentation – could address this in the future. Another challenge comes while evaluating the accuracy of parses; our framework depends on an exact match to a logical form for a sentence parsed by a human, which is an overly strict criterion. This is a problem that also plagues other approaches such as fully-supervised syntactic parsing (Berzak et al., 2016). Two logical forms may express the same meaning but be written in different ways. For example, if the sentence is *The person walked toward the table*, two different yet equally valid logical forms are

$$\text{(1) person}(x) \wedge \text{walk}(x) \wedge \text{toward}(x, y) \wedge \text{table}(y)$$
$$\text{(2) person}(x) \wedge \text{approach}(x, y) \wedge \text{table}(y)$$

However, these logical forms differ significantly (differences shown in teal above). The only overlapping predicates are the objects and, even using our near miss criterion, these sentences still fail because they differ by more than a single predicate. Working with videos can be challenging in general due to the sheer scale of information. In order to efficiently encode the videos, we needed to pretrain the visual system and make thoughtful choices like caching and evaluating sub-expressions for early stopping of invalid predictions. This worked in our framework due to having an intentionally smaller dataset, but would have been difficult with large-scale datasets currently used for most VL models.

In the second approach, we move into an interactive domain where we train a semantic parser grounded in a robotic environment. We paired highly temporal, constrained natural language with a robotic planner. The semantic parser hypothesizes executable formulas for the commands and verifies them using the planner, which serves as the supervision. The formalism, linear temporal logic (LTL), allows for more representational power in terms of what concepts could be represented.

Much like the first semantic parser presented in this thesis, we do not rely on providing the model extensive prior knowledge about the formalism. This approach actually has even less prior knowledge in that it is not provided with a seed lexicon and instead only knows about well-formedness. Even without any built-in knowledge of LTL, the model was able to learn the syntactic rules and semantic categories of the formalism. The learned knowledge about LTL was enough to generate accurate formulas to be used downstream in the robotic simulator. We again release the dataset of both grammar-generated and human-generated sentences paired with multiple execution sequences.

We explored both grammar- and human-generated sentences in this work as well. To get human-generated sentences, annotators were shown three execution traces and asked to write a command. We did note that some commands did not fully capture the “ground-truth" meaning as represented by the LTL formula because three execution traces is not enough to enumerate all possibilities. Nonetheless, the human generated sentences had far fewer constraints and appeared to more efficiently communicate the intent given their shorter length on average. The grammar-generated commands, on the other hand, were more systematic, using a smaller vocabulary. We do again note that one difficulty directly comparing the two groups of sentences is that we do not have a guarantee of exactly equivalent semantic meaning. Nonetheless, most commands did a good job of capturing the meaning as evidenced by the high performance on human commands that is compared to the grammar-generated commands. We hope this publicly available dataset will be useful for other researchers engaged in using the LTL formalism in a robotic setting. In general, the ability to process and comprehend temporal language is a challenging one yet largely important as we move toward language models for robots that interact with humans.

In both of the above works, we places a large emphasis on natural language generated by human annotators. We wanted to move beyond just relying on a grammar to generate our sentences. While grammars are systematic and guaranteed to refer to the target content (take for instance the commands generated directly from the LTL formulas versus the commands from annotators just looking at the demonstrations), they do not have the linguistic diversity seen in unconstrained language from humans. As we are interested in models of child language acquisition, going a step further and using child-directed speech would be better. Abend et al. (2017) used child-directed speech from the Eve dataset (Brown and Bellugi,

1964) to train the semantic parser, though their approach only used natural language as input. Corpora like CHILDES (MacWhinney, 2000) contain large amounts of annotated, child-directed speech; however the data are not often couples with perceptual information. Even videos are usually of interviews or children interacting with adults, so this does still pose a challenge about how to acquire large scale, clean child-directed speech. In future work, we aim to focus on directing annotators to provide sentences as if they were speaking to a child as a soft proxy for child-directed speech.

We next focused on how the representations learned by multimodal models are used for generalization from two angles. From the first angle, we looked at systematic generalization using the gSCAN benchmark. We use the pretrain-then-finetune approach to focus on flexible, reusable linguistic representations inspired by the flexibilty seen by children during linguistic generalizations. Our model is a Transformer-based encoder-decoder model that uses a novel decoding strategy and has demonstrable improvements on the gSCAN benchmark. The encoder and decoder are each separately pretrained and then jointly finetuned on the gSCAN task of mapping natural language commands and an initial world state to a sequence of actions. For the other angle, we focused on one tangible generalization failure by models, which is that they learn social biases like children and adults yet fail to generalize beyond these biases in light of new data. We contribute new metrics, Grounded-WEAT and Grounded-SEAT, that are adapted from prior metrics in psychology and NLP. we use these metrics address three specific questions about pretrained vision and language Transformers. Namely, we find that vision and language Transformers do encode social bias, are not robust to adjusting to new counterstereotypical evidence, and that the bias generally comes from both language and vision.

Alongside these metrics, we also release a dataset of images paired with words and sentences across 13 different tests for social bias. These web-scraped images are human-verified and represent more diversity among people than found in existing image captioning datasets. Using GWEAT and GSEAT, we find that visually grounded word embeddings do in fact encode social biases, that counter stereotypical grounded evidence has little impact on social biases, and that biases largely come from language, rather than being introduced by vision. We probe four V&L Transformer – two single stream and two dual stream;

in future work, we intend to test GWEAT/GSEAT on more VL models as well. It may be that models like ViLT (Kim et al., 2021), which uses image patches instead of region features from a CNN, are less biased or perhaps encode biases different; a pretraining approach like VILLA (Gan et al., 2020) that uses adversarial noise to perturb images during pretraining may be fairer in their visual features. A more in-depth, systematic study across architectures, pretraining approaches and pretraining datasets is a natural next step based on this work. Also, as more vision-and-language Transformer models are released, we hope to see researchers utilizing these metrics as a benchmark for the implicit prejudices being learned. Additionally, while we focused on vision and language, many other modalities can and should be tested, such as audio and language embeddings (Kiela and Clark, 2015; Torabi et al., 2016; Iashin and Rahtu, 2020). There are also known disparities in performance in areas such a voice recognition (Tatman, 2017), making an audio a particularly interesting next direction. We note again this is nontrivial as it does require a dataset that can specifically probe biases, but the metrics we presented here can be used out-of-the-box.

Bibliography

Omri Abend, Tom Kwiatkowski, Nathaniel J Smith, Sharon Goldwater, and Mark Steedman. 2017. Bootstrapping language acquisition. *Cognition* 164:116–143.

Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. 2019. Learning to Generalize from Sparse and Underspecified Rewards. In *Proceedings of the International Conference on Machine Learning (ICML)*. pages 130–140.

Muhannad Al-Omari, Paul Duckworth, David C Hogg, and Anthony G Cohn. 2017. Natural Language Acquisition and Grounding for Embodied Robotic Systems. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*. pages 4349–4356.

Elaine S. Andersen, Anne Dunlea, and Linda Kekelis. 1993. The impact of input: language acquisition in the visually impaired. *First Language* 13(37):23–49.

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*.

Jacob Andreas, Dan Klein, and Sergey Levine. 2017. Modular Multitask Reinforcement Learning with Policy Sketches. In *Proceedings of the International Conference on Machine Learning (ICML)*. JMLR. org, pages 166–175.

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic Parsing as Machine Translation. In *Proceedings of the Conference for Association for Computational Linguistics*. pages 47–52.

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 (ICCV)*. pages 2425–2433.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neurla Machine Translation. *International Conference on Learning Representations (ICLR)* .

Yoav Artzi. 2016. Cornell SPF: Cornell Semantic Parsing Framework.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the Association for Computational Linguistics (TACL)* 1:49–62.

Dzmitry Bahdanau, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. 2019a. CLOSURE: Assessing Systematic Generalization of CLEVR Models. *arXiv preprint arXiv:1912.05783* .

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019b. Systematic Generalization: What Is Required and Can It Be Learned? In *International Conference on Learning Representations (ICLR)*.

David Barner and Jesse Snedeker. 2008. Compositionality and Statistics in Adjective Acquisition: 4-Year-Olds Interpret *Tall* and *Short* Based on the Size Distributions of Novel Noun Referents. *Child Development* 79(3):594–608.

Andrew Scott Baron and Mahzarin R Banaji. 2006. The Development of Implicit Attitudes. *Psychological Science* 17(1):53–58.

Jasmijn Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. 2018. Jump to better conclusions: Scan both left and right. In *BlackboxNLP@EMNLP*.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1533–1544.

Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. pages 1415–1425.

Marianne Bertrand and Sendhil Mullainathan. 2004. Are Emily and Greg More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. *American Economic Review* 94(4):991–1013.

Yevgeni Berzak, Andrei Barbu, Daniel Harari, and Boris Katz. 2015. Do You See What I Mean? Visual Resolution of Linguistic Ambiguities. *Conference on Empirical Methods on Natural Language Processing (EMNLP)* pages 1477–1487.

Yevgeni Berzak, Yan Huang, Andrei Barbu, Anna Korhonen, and Boris Katz. 2016. Anchoring and Agreement in Syntactic Annotations. *Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

Ann Bigelow. 1987. Early words of blind children. *Journal of child language* 14(1):47–56.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience Grounds Language. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics (ACL), pages 8718–8735.

Irene V Blair, Jennifer E Ma, and Alison P Lenton. 2001. Imagining stereotypes away: the moderation of implicit stereotypes through mental imagery. *Journal of Personality and Social Psychology* 81(5):828.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hann Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*. pages 4349–4357.

Martin DS Braine. 1971. On two types of models of the internalization of grammars. *The Ontogenesis of Grammar* pages 153–186.

Roger Brown and Ursula Bellugi. 1964. Three Processes in the Child's Acquisition of Syntax. *Harvard Educational Review* 34(2):133–151.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bob Carpenter. 1997. *Type Logical Semantics*. MIT press.

David L. Chen and Raymond J. Mooney. 2011. Learning to Interpret Natural Language Navigation Instructions from Observations. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*. San Francisco, CA, USA.

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:1504.00325* .

Yen-Chun Chen, Linjie Li, Licheng Yu, A. E. Kholy, Faisal Ahmed, Zhe Gan, Y. Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Noam Chomsky. 1965. Aspects of the Theory of Syntax.

Michelle M Chouinard and Eve V Clark. 2003. Adult Reformulations of Child Errors as Negative Evidence. *Journal of Child Language* 30(3):637–669.

Patricia Hill Collins. 2004. *Black Sexual Politics: African Americans, Gender, and the New Racism*. Routledge.

Corey Columb and E Ashby Plant. 2016. The Obama Effect Six Years Later: The Effect of Exposure to Obama on Implicit Anti-Black Evaluative Bias and Implicit Racial Stereotyping. *Social Cognition* 34(6):523–543.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically Motivated Large-Scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, Prague, Czech Republic, pages 33–36.

Dario Cvencek, Anthony G. Greenwald, and Andrew N. Meltzoff. 2011. Measuring implicit attitudes of 4-year-olds: the preschool implicit association test. *Journal of Experimental Child Psychology* 109(2):187–200.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics, Minneapolis, Minnesota.

Li Dong and Mirella Lapata. 2016. Language to Logical Form with Neural Attention. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. Berlin, Germany, pages 33–43.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, Analogies, and Machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING, the International Conference on Computational Linguistics*. pages 3519–3530.

Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Etienne Renault, and Laurent Xu. 2016. Spot 2.0—A Framework for LTL and ω-Automata Manipulation. In *International Symposium on Automated Technology for Verification and Analysis*. Springer, pages 122–129.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* .

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving Text-to-SQL Evaluation Methodology. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics (ACL), pages 351–360.

Michael C Frank and Noah D Goodman. 2012. Predicting Pragmatic Reasoning in Language Games. *Science* 336(6084):998–998.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *ArXiv* abs/2006.06195.

Tong Gao, Qi Huang, and Raymond J Mooney. 2020. Systematic Generalization on gSCAN with Language Conditioned Embedding. In *Proceedings of the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL)*. Association of Computational Linguistics.

Dedre Gentner. 1982. Why Nouns are Learned Before Verbs: Linguistic Relativity versus Natural Partitioning. *Center for the Study of Reading Technical Report; no. 257* .

Samuel J. Gershman and Joshua B. Tenenbaum. 2015. Phrase similarity in humans and machines. *Cognitive Science* .

Jane Gillette, Henry Gleitman, Lila Gleitman, and Anne Lederer. 1999. Human simulations of vocabulary learning. *Cognition* 73(2):135–176.

Lila R. Gleitman and Henry Gleitman. 1992. A Picture Is Worth a Thousand Words, but That's the Problem: The Role of Syntax in Vocabulary Acquisition. *Current Directions in Psychological Science* 1:31 – 35.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, pages 650–655.

Omer Goldman, Veronica Latcinnik, Ehud Nave, Amir Globerson, and Jonathan Berant. 2018. Weakly Supervised Semantic Parsing with Abstract Examples. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics (ACL), Melbourne, Australia, pages 1809–1819.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC)*.

Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of Personality and Social Psychology* 74(6):1464.

Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. 2017. From Language to Programs: Bridging Reinforcement Learning and Maximum Marginal Likelihood. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. pages 1051–1062.

Stevan Harnad. 2007. Symbol Grounding Problem. *Scholarpedia* 2:2373.

Betty Hart and Todd R Risley. 2003. The Early Catastrophe. The 30 Million Word Gap. *The American Educator* 17(1).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 770–778.

Christina Heinze-Deml and Diane Bouchacourt. 2020. Think before you act: A simple baseline for compositional generalization. *arXiv preprint arXiv:2009.13962* .

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. 2017. Grounded Language Learning in a Simulated 3D World. *arXiv* .

Satoshi Horie and Osamu Watanabe. 1997. Hard instance generation for SAT. In *International Symposium on Algorithms and Computation*. Springer, pages 22–31.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative Answer Prediction With Pointer-Augmented Multimodal Transformers for TextVQA. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 9989–9999.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In *FINDINGS*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality Decomposed: How doNeural Networks Generalise? *Journal of Artificial Intelligence Research (JAIR)* 67:757–795.

Vladimir Iashin and Esa Rahtu. 2020. A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer. In *British Machine Vision Conference (BMVC)*.

Rohan Jha, Charles Lovering, and Ellie Pavlick. 2020. When does data augmentation help generalization in NLP? *ArXiv* .

Robin Jia and Percy Liang. 2016. Data Recombination for Neural Semantic Parsing. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Berlin, Germany, pages 12–22.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 2901–2910.

Tim Johnson. 1984. Natural Language Computing: The Commercial Applications. *The Knowledge Engineering Review* 1(3):11–23.

Douwe Kiela and Stephen Clark. 2015. Multi- and Cross-Modal Semantics Beyond Vision: Grounding in Auditory Perception. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 2461–2470.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. *ArXiv* abs/2010.05465.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*. PMLR, pages 5583–5594.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Tom Kocmi and Ondřej Bojar. 2017. An Exploration of Word Embedding Initialization in Deep-Learning Tasks. *Proceedings of the International Conference on Natural Language Processing (ICON)* .

Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.

Yen-Ling Kuo, Andrei Barbu, and Boris Katz. 2018. Deep Sequential Models for Sampling-Based Planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pages 6490–6497.

Yen-Ling Kuo, Boris Katz, and Andrei Barbu. 2020a. Compositional Networks Enable Systematic Generalization for Grounded Language Understanding. In *Findings on Empirical Methods in Natural Language Processing (EMNLP)*.

Yen-Ling Kuo, Boris Katz, and Andrei Barbu. 2020b. Encoding formulas as deep networks: Reinforcement learning for zero-shot execution of LTL formulas. In *International Conference on Intelligent Robots and Systems (IROS)*. pages 5604–5610.

Tom Kwiatkowksi, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1223–1233.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling Semantic Parsers with On-the-Fly Ontology Matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1545–1556.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical Generalization in CCG Grammar Induction for Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1512–1523.

Brenden Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Brenden M. Lake and Gregory L. Murphy. 2021. Word meaning in minds and machines. *Psychological review* .

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.

Barbara Landau, Lila R Gleitman, and Barbara Landau. 2009. *Language and experience: Evidence from the blind child*, volume 8. Harvard University Press.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*. volume 34, pages 11336–11344.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv:1908.03557* .

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020b. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth Forbus, and Ni Lao. 2017. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. pages 23–33.

L. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing Across Time: What Does RoBERTa Know and When? In *EMNLP*.

Qi Liu, Matt J Kusner, and Phil Blunsom. 2020. A Survey on Contextual Embeddings. *arXiv* .

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* .

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*. pages 13–23.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-Task Vision and Language Representation Learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. pages 10437–10446.

Brian MacWhinney. 2000. *The CHILDES Project. Volume II: The Database*, volume 2. Psychology Press.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. pages 1671–1678.

Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bryan McCann, James Bradbury, Caiming Xiong, and R. Socher. 2017. Learned in Translation: Contextualized Word Vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Colleen McDonough, Lulu Song, Kathy Hirsh-Pasek, Roberta Michnick Golinkoff, and Robert P. Lannon. 2011. An image is worth a thousand words: why nouns tend to dominate verbs in early word learning. *Developmental Science* 14 2:181–199.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*. pages 3111–3119.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR), pages 1928–1937.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations*. CONF.

Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2019. Large-scale Pretraining for Visual Dialog: A Simple State-of-the-Art Baseline. In *European Conference on Computer Vision (EECV)*.

Roma Patel, Roma Pavlick, and Stefanie Tellex. 2019. Learning to Ground Language to Temporal Logical Form. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. SpLU RoboNLP Workshop.

Tabitha C Peck, Sofia Seinfeld, Salvatore M Aglioti, and Mel Slater. 2013. Putting yourself in the skin of a black avatar reduces implicit racial bias. *Consciousness and Cognition* 22(3):779–787.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Steven Piantadosi and Richard Aslin. 2016. Compositional Reasoning in Early Childhood. *PloS one* 11(9).

Amy E Pierce. 1992. Language acquisition and syntactic theory. In *Language Acquisition and Syntactic Theory*, Springer, pages 1–17.

Steven Pinker. 1989. The Acquisition of Argument Structure. In *Language, Cognition, and Human Nature*, Oxford University Press.

Amir Pnueli. 1977. The temporal logic of programs. In *Annual Symposium on Foundations of Computer Science*. IEEE, pages 46–57.

Z Manna A Pnueli and Z Manna. 1990. A hierarchy of temporal properties. *Proceedings of the ACM Symposium on Principles of Distributed Computing* .

Linlu Qiu, Hexiang Hu, Bowen Zhang, Peter Shaw, and Fei Sha. 2021. Systematic Generalization on gSCAN: What is Nearly Solved and What is Next? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI preprint* .

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research (JMLR)* pages 1–67.

Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anna Rogers. 2021. Changing the world by changing the data. *arXiv preprint arXiv:2105.13947* .

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics* 8:842–866.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics* 8:264–280.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and B. Lake. 2020. A Benchmark for Systematic Generalization in Grounded Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Laura Ruis and Brenden Lake. 2022. Improving systematic generalization through modularity and augmentation. *arXiv preprint arXiv:2202.10745* .

Luana Ruiz, Joshua Ainslie, and Santiago Ontañón. 2021. Iterative Decoding for Compositional Generalization in Transformers. *arXiv preprint arXiv:2110.04169* .

Matthew Setzler, Scott Howland, and Lauren Phillips. 2022. Recursive decoding: A situated cognition approach to compositional generation in grounded language understanding. *arXiv preprint arXiv:2201.11766* .

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*.

Narayanaswamy Siddharth, Andrei Barbu, and Jeffrey Mark Siskind. 2014. Seeing What You're Told: Sentence-Guided Activity Recognition In Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computing Research Repository (CoRR)* .

Tejas Srinivasan and Yonatan Bisk. 2021. Worst of Both Worlds: Biases Compound in Pre-trained Vision-and-Language Models. *arXiv preprint arXiv:2104.08666* .

Ryan Steed and Aylin Caliskan. 2021. Image Representations Learned With Unsupervised Pre-Training Contain Human-like Biases. *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency* .

Mark Steedman. 1996. *Surface Structure and Interpretation*, volume 1. The MIT Press.

Mark Steedman. 2000. *The Syntactic Process*, volume 1. The MIT Press.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *International Conference on Learning Representations (ICLR)*.

Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Haochen Tan and Mohit Bansal. 2020. Vokenization: Improving Language Understanding via Contextualized, Visually-Grounded Supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yi Chern Tan and L Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. In *Advances in Neural Information Processing Systems (NeurIPS)*. pages 13209–13220.

Lappoon R. Tang and Raymond J. Mooney. 2001. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In *European Conference on Machine Learning*. pages 466–477.

Rachael Tatman. 2017. Gender and Dialect Bias in YouTube's Automatic Captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*.

Atousa Torabi, Niket Tandon, and Leonid Sigal. 2016. Learning Language-Visual Embedding for Movie Understanding with Natural-Language. *arXiv:1609.08124* .

Ramakrishna Vedantam, Arthur Szlam, Maximillian Nickel, Ari Morcos, and Brenden M Lake. 2021. CURI: A Benchmark for Productive Concept Learning under Uncertainty . In *International Conference on Machine Learning (ICLR)*. Proceedings of Machine Learning Research (PMLR), pages 10519–10529.

Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL*.

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. 2019. Balanced Datasets Are Not Enough: Estimating and Mitigating Gender Bias in Deep Image Representations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pages 5310–5319.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R Bowman. 2020. Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Terry Winograd. 1971. Procedures As A Representation For Data In A Computer Program For Understanding Natural Language. Technical report, Massachusetts Institute of Technology.

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

William Woods. 1973. Progress in Natural Language Understanding: An Application to Lunar Geology. In *Proceedings of the National Computer Conference and Exposition*. Association for Computing Machinery (ACM), New York, NY, USA, page 441–450.

Haonan Yu, N. Siddharth, Andrei Barbu, and Jeffrey Mark Siskind. 2015. A Compositional Framework for Grounding Language Inference, Generation, and Acquisition in Video. *Journal of Artificial Intelligence Research (JAIR)* .

Tian Yun, Chen Sun, and Ellie Pavlick. 2021. Does Vision-and-Language Pretraining Improve Lexical Grounding? *ArXiv* abs/2109.10246.

John M Zelle and Raymond J Mooney. 1996. Learning to Parse Database Queries Using Inductive Logic Programming. In *Proceedings of the National Conference on Artificial Intelligence*. page 1050–1055.

Luke Zettlemoyer and Michael Collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. pages 658–666.

Luke S Zettlemoyer and Michael Collins. 2007. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 678–687.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021a. Vinvl: Revisiting visual representations in vision-language models. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pages 5575–5584.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R Bowman. 2021b. When Do You Need Billions of Words of Pretraining Data? In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender Bias in Contextualized Word Embeddings. In *Proceedings of the Conference for the Association for Computational Linguistics (ACL)*. Minneapolis, Minnesota, pages 629–634.

List of Figures

List of Tables

Appendix A

Appendix

Seed Lexicon for CCG

Below, we show the seed lexicon used for in Chapter 2. This seed lexicon was created by randomly sampling 26 sentences (2% of all sentences in the dataset), manually parsing these sentences, and adding the lexical entries they contain to the seed. These sentences were excluded from the training and evaluation data.

token	*syntactic tag*	*semantic tag*
she	NP	$\lambda\, x.\, person(x)$
put	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\ put\_down(x,y) \wedge f(x) \wedge g(y)$
the	NP/N	$\lambda fx.\ f(x)$
orange	N/N	$\lambda\ fx.\ orange(x) \wedge f(x)$
car	N	car
on	(NP\NP)/NP	$\lambda w.\lambda x \wedge on(y,z)$
desk	N	$table$
there is	S/NP	$\lambda f.\lambda x.\lambda y.\ f(x,y)$
a	NP/N	$\lambda fx.\ f(x)$
green	N/N	$\lambda\ fx.\ green(x) \wedge f(x)$
backpack	N	bag
on	(NP\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.on(x,y) \wedge f(x) \wedge g(y)$

yellow	N/N	$\lambda\ fx.\ yellow(x) \wedge f(x)$
chair	N	*chair*
both	NP/N	$\lambda f.\lambda x\ two(x) \wedge f(x)$
guys	N	*person*
set down	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\ put\_down(x,y) \wedge f(x) \wedge g(y)$
man	N	*person*
is	(S\NP)/(S\NP)	$\lambda f.\lambda g.\lambda x.\lambda y\ f(x) \wedge g(x)$
is	((S\NP)/NP)/NP	$\lambda f.\lambda g.\lambda h.\lambda x.\lambda y.\lambda z f(x,y) \wedge g(y) \wedge h(x,z)$
near	PP/NP	$\lambda f.\ \lambda x.\ \lambda y\ near(x,y) \wedge f(x)$
on	PP/NP	$\lambda f.\ \lambda x.\ \lambda y\ on(x,y) \wedge f(x)$
woman	N	*person*
grabs	(S\NP)/NP	$\lambda f.\ \lambda x.\ \lambda y pick\_up(x,y) \wedge f(x)$
an	NP/N	$\lambda fx.\ f(x)$
apple	N	*apple*
off	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.from(x,y) \wedge f(x) \wedge g(y)$
table	N	*table*
guy	N	*person*
in	(NP\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y\ f(x) \wedge g(x) \wedge in(x,y)$
plaid	N/N	$\lambda\ fx.\ plaid(x) \wedge f(x)$
shirt	N	*shirt*
walks	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\lambda z\ walk(x) \wedge f(x,y) \wedge g(x,z)$
stands	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\lambda z\ stand(x) \wedge f(x,y) \wedge g(x,z)$
towards	PP/NP	$\lambda f.\ \lambda x.\ \lambda y\ toward(x,y) \wedge f(x)$
tan	N/N	$\lambda\ fx.\ tan(x) \wedge f(x)$
is	(S\NP)/(S\NP)	$\lambda f.\lambda g.\lambda x.\lambda y\ f(x,y) \wedge g(x)$
standing	S\NP	$\lambda x\ stand(x)$
by	(S\NP)/(S\NP)	$\lambda f.\lambda g.\lambda x.\lambda y.\ near(x,y) \wedge f(x) \wedge g(x)$
lifting	(NP\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\ f(x) \wedge g(x) \wedge pick\_up(x,y)$
is	(S\NP)/(S\NP)	$\lambda f.\lambda g.\lambda x.\lambda y.\lambda z.f(x,y) \wedge g(x,z)$
wearing	(S\NP)/NP	$\lambda f.\ \lambda x.\ \lambda y wear(x,y) \wedge f(x)$

walks	(S\NP)/NP	$\lambda f.\ \lambda g.\lambda h.\lambda x.\ \lambda y.\lambda z.\ walk(x) \wedge f(x) \wedge g(y,z)$
over to	(NP\NP)/NP	$\lambda f.\ \lambda g.\lambda x.\ \lambda y.\ to(x,y) \wedge f(x) \wedge g(y)$
another	NP/N	$\lambda fx.\ f(x)$
blue	N/N	$\lambda\ fx.\ blue(x) \wedge f(x)$
jacket	N	*shirt*
staring at	(S\NP)/NP	$\lambda f.\ \lambda x.\ \lambda ylook\_at(x,y) \wedge f(x)$
office chair	N	*chair*
picks up	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\lambda z.\ pick\_up(x,z) \wedge f(x,y) \wedge g(z)$
is	(S\NP)/NP	$\lambda f.\lambda x.\ f(x), g(x)$
red	N/N	$\lambda\ fx.\ red(x) \wedge f(x)$
there is	S/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\lambda z.\ f(x,y) \wedge g(x,z)$
in the middle of	(NP\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda y.\ in(x,y) \wedge f(x) \wedge g(y)$
lobby	N	*lobby*
picking up	PP/NP	$\lambda f.\ \lambda x.\ \lambda y\ pick\_up(x,y) \wedge f(x)$
is	((S\NP)/NP)/NP	$\lambda f.\lambda g.\lambda h.\lambda w.\lambda x.\lambda y.\lambda z.\ f(w,x) \wedge g(w,z) \wedge h(y,z)$
sliding	PP/NP	$\lambda f.\ \lambda x.\ \lambda y\ move(x,y) \wedge f(x)$
to	PP/NP	$\lambda f.\lambda g.\lambda x.\lambda yto(x,y) \wedge f(x,y)$
his	NP/N	$\lambda fx.\ f(x)$
right	N	*right*
placed	S\NP	$\lambda f.\lambda x.\ put\_down(x)$
on	((S\NP)\(S\NP))/NP	$\lambda f.\lambda g.\lambda w.\lambda x.\lambda y.\ f(x) \wedge g(x) \wedge on(x,y)$
gives	((S\NP)/NP)/NP	$\lambda f.\lambda g.\lambda h.\lambda x.\lambda y.\lambda z.\ f(x) \wedge g(y) \wedge h(z) \wedge give(x,y,z)$
pear	N	*pear*
grey	N/N	$\lambda\ fx.\ gray(x) \wedge f(x)$
walks	((S\NP)/NP)/NP	$\lambda f.\lambda g.\lambda h.\lambda w.\lambda x.\lambda y.\lambda z.\ f(w,y) \wedge g(y,z) \wedge h(w,x) \wedge walk(w)$
away from	PP/NP	$\lambda f.\lambda x.\lambda y.\ f(y) \wedge from(x,y)$
hold	PP/NP	$\lambda f.\lambda x.\lambda y.\ f(x) \wedge hold(x,y)$
moves	(S\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\lambda z.\ f(x) \wedge g(y,z) \wedge move(x,y)$

book	N	*book*
across	(NP\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\ f(x) \wedge g(y) \wedge across(x, y)$
sets	(S\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\lambda z.\ f(x) \wedge g(y, z) \wedge put\_down(x, y)$
scope	N	*telescope*
down on	(NP\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\ f(x) \wedge g(y) \wedge on(x, y)$
one	NP/N	$\lambda f.\lambda x\ two(x) \wedge f(x)$
places	(S\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\lambda z.\ f(x) \wedge g(y, z) \wedge put\_down(x, y)$
holds out	(S\NP)/NP	$\lambda f.\lambda g.\lambda x.\lambda x.\lambda y.\ f(x) \wedge g(y) \wedge hold(x, y)$
swinging	(S\NP)/NP	$\lambda f.\lambda x.\lambda x.\lambda y.\ f(y) \wedge move(x, y)$
two	NP/N	$\lambda f.\lambda x\ two(x) \wedge f(x)$
2	NP/N	$\lambda f.\lambda x\ two(x) \wedge f(x)$

Grammar for Generating Natural Language Commands Using LTL Formalism

Below, we detail the grammar described in Section 3.5 used to generated the natural language commands for training the LTL-based parser.

BINOP → ‘and’ | ‘or’

UOP → ‘do not’ | ‘you should not’

ITEM → ‘apple’ | ‘orange’ | ‘pear’

LANDMARK → ‘flag’ | ‘house’ | ‘tree’

PREDICATE → ‘be around the’ LANDMARK | ‘be near the’ LANDMARK

| ‘go to the’ LANDMARK | ‘hold the’ Item

| ‘take the’ ITEM | ‘possess the’ ITEM

P → PREDICATE | UOP PREDICATE | PREDICATE BINOP PREDICATE | UOP P

S → SAFETY | GUARANTEE | OBLIGATION | RECURRENCE |

| PERSISTENCE | REACTIVITY

SPREFIX → ‘always’ | ‘at all times,’

SSUFFIX → ‘forever’ | ‘at all times’ | ‘all the time’

SAFETY → SPREFIX P | P SSUFFIX | SAFETY BINOP SAFETY

GPREFIX → ‘eventually’ | ‘at some point’

NOTPREDICATE → UOP PREDICATE

GUARANTEE → GPREFIX P | ‘guarantee that you will’ PREDICATE

| ‘guarantee that you’NOTPREDICATE | GUARANTEE BINOP GUARANTEE

OBLIGATION → SAFETY BINOP GUARANTEE | OBLIGATION BINOP SAFETY

| OBLIGATION BINOP GUARANTEE

RECURRENCE → ‘eventually,’ P ‘and do this repeatedly’ | RECURRENCE BINOP RECURRENCE

PERSISTENCE → ‘at some point, start to’ P ‘and keep doing it’

| PERSISTENCE BINOP PERSISTENCE

REACTIVITY → RECURRENCE BINOP PERSISTENCE | REACTIVITY BINOP RECURRENCE

| REACTIVITY BINOP PERSISTENCE

Training & Inference Splits for gSCAN

GSCAN has a *compositional split* training set with 8 rule-based generalization splits and a single *length* training set with a single length generalization split. We detail the generalization splits below.

A: Random – For this split, examples from the training and test set are drawn from the same distribution. There are not any systematic differences between the training and test splits, which provides a baseline for model performance when no systematic generalization is required.

B: Yellow Squares – This split tests how well the model can interpret novel combinations of attributes. During training, whenever a yellow square is the target object, the color *yellow* is never mentioned. The square can be referred to by its size or shape only. During inference, the model must learn to ground the two known concepts of *yellow* and *square* together.

C: Red Squares – This split is a more difficult version split B in testing the model's ability to compose attributes. Red squares appear in training as distractor objects only and are never mentioned as the target object. During inference, the model must generalize that red squares can be also be targets.

D: Novel Direction – In this split, during training there are no examples where the target object is south-west of the agent. The model must generalize being able to walk in this direction to the target during inference.

E: Relativity – This split tests a model's ability to understand that linguistic concepts like size are relative to the other objects in the grid. During training, objects of a given size are never referred to as *small* and only by other attributes such as their color or shape. During inference, the model must generalize that objects of this size can be referred to as small based on the other objects around them.

F: Class Inference – For this split, all examples where heavy objects of size 3 are pushed are held out during training. The model must infer these objects are heavy for objects and therefore requiring two pushes from examples of them being pulled only.

G: Adverb-k – This split tests the model's few shot generalization ability. Of all of the examples using the adverb *cautiously*, *k* examples are randomly selected to go in the training set. The remaining examples are used for inference.

H: Adverb to Verb – This split tests the models ability to generalize the adverb *while spinning* to the verb *pull* because during training, *while spinning* is only ever used with the verb *push*.

I: Novel Length – All examples in the length training set are up to but not longer than 15 target actions. During inference, the model must learn to generalize to commands requiring longer action sequences. This split also uses a larger grid (12×12 cells compared to 6×6 cells used by the previous train/inference splits).