Improving Coherence and Consistency in Neural Sequence Models with Dual-System, Neuro-Symbolic Reasoning

Maxwell Nye[1*], **Michael Henry Tessler**[1], **Joshua B. Tenenbaum**[1], **Brenden M. Lake**[2]
[1]MIT [2]NYU and Facebook AI Research

Abstract

Human reasoning can often be understood as an interplay between two systems: the intuitive and associative ("System 1") and the deliberative and logical ("System 2"). Neural sequence models—which have been increasingly successful at performing complex, structured tasks—exhibit the advantages and failure modes of System 1: they are fast and learn patterns from data, but are often inconsistent and incoherent. In this work, we seek a lightweight, training-free means of improving existing System 1-like sequence models by adding System 2-inspired logical reasoning. We explore several variations on this theme in which candidate generations from a neural sequence model are examined for logical consistency by a symbolic reasoning module, which can either accept or reject the generations. Our approach uses neural inference to mediate between the neural System 1 and the logical System 2. Results in robust story generation and grounded instruction-following show that this approach can increase the coherence and accuracy of neurally-based generations.

1 Introduction

Despite recent success, neural sequence models often fail to produce consistent and coherent generations. When generating stories, language models may forget the attributes of specific characters (such as personality and background information) (Welleck et al., 2018), ignore previously established relationships between characters (such as family relationships) (Sinha et al., 2019), or otherwise contradict prior statements (Brown et al., 2020). Similarly, neural models can make statements that contradict basic world knowledge or the logical entailment structure of known facts.

Lake & Murphy (2020) illustrated several of these issues with GPT-2 (Radford et al., 2019). When given prompts of the form *"A dolphin is a ___"*, GPT-2 predicts that the most likely answer is "mammal", "fish", or "bird" depending on small differences in the wording of the prompt. In another example, GPT-2 states that unicorns have "four horns," directly after implying that unicorns only have one horn. Upon diagnosing such issues, it is unclear how to apply a targeted fix to the model, especially if retraining or fine-tuning is impractical.

In this work, we draw on insights from cognitive science, especially from "dual process" theories of reasoning (Evans, 2003), to explore how neural sequence models can better interface with prior knowledge and be made more coherent and consistent. According to dual process theories, human cognition can be understood as an interplay between a more intuitive and associative "System 1" and a more deliberative and logical "System 2." Within this broad framework, automatic actions are driven by System 1, whereas System 2 engages for more deliberative control: for example, judging the validity of a logical argument that requires multiple steps of reasoning (Kahneman, 2013).

The prominent neural language models of today are single systems, with weaknesses akin to those exhibited by the human System 1. For example, the cognitive reflection test (CRT) (Frederick, 2005) is a classic probe of System 1 vs. System 2 reasoning in humans. Participants answer a set of simple questions that have superficially compelling, but logically invalid, answers. These incorrect answers are often generated as a first "gut" response (putatively, by System 1 intuitive thinking); upon reflection,

[*] Correspondence to mnye@mit.edu. Work primarily done during an internship at Facebook AI Research.

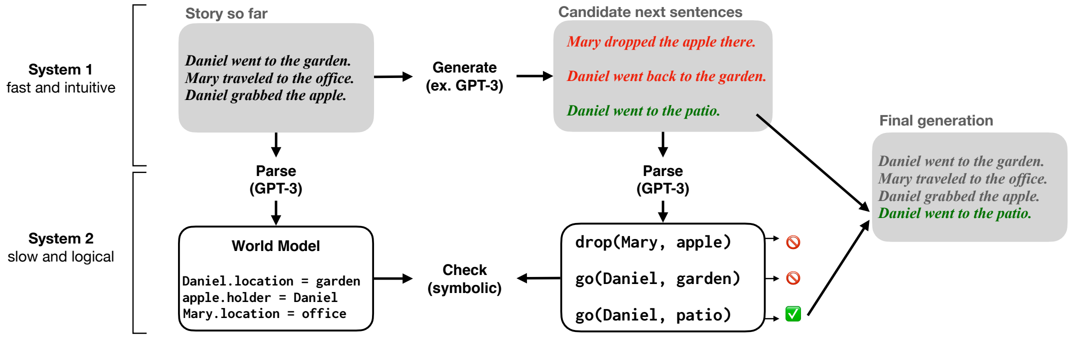


Figure 1: Schematic of dual-system approach to text generation. Conditioned on previous text, a "System 1" neural generation model produces candidate next sentences. Semantic parses for each candidate are generated via few-shot parsing from GPT-3 and compared to a minimal world model to check consistency. Only candidates consistent with the world model state are incorporated into the final generation.

however, participants often realize that their responses were not logically or mathematically consistent (via more explicit System 2 reasoning). Consider the CRT problem on the left below:

*A ball and a bat cost **$1.10**.*
The bat costs one dollar more than the ball.
How much does the ball cost?

Total cost in prompt	GPT-3 response
$1.10	10 cents
$1.20	20 cents
$1.30	$0.30
$1.70	$0.70

Reading quickly, you might be tempted to say the ball costs 10 cents. Most participants give this response, in fact, especially if they are under time pressure or have limited attention (Kahneman, 2013). Of course, if the bat is $1.00 more than the ball, and the ball costs 10 cents, then the *total* cost would be $1.20. The correct answer is that the ball costs 5 cents. Notably, in this and other classic CRT problems, GPT-3 (Brown et al., 2020) predicts the same "gut" response (prediction in red above; the table above shows that adjusting the price in the prompt also leads to similar effects; see Appendix Figure 8 for more CRT examples). GPT-3 appears vulnerable to the same sort of intuitive, unsystematic pattern recognition errors as humans—in this case, incorrectly subtracting one dollar from $1.10, without confirming that the answer satisfies each of the problem constraints.

Numerous studies have shown that engagement of System 2-style effort can help "override or inhibit default responses emanating from System 1" (Evans, 2003), correcting inconsistent or un-systematic intuitive impulses. For example, when System 2 is engaged by asking people to take more time to respond, people's accuracy improves on the CRT task above (Kahneman, 2013). It has been argued that integrating System 2 processing could similarly improve AI systems (Goyal & Bengio, 2020; Garcez & Lamb, 2020), and here we explore this idea as applied to neural sequence models.

In this work, we take inspiration from dual process theories to explore a neuro-symbolic generation system, wherein predictions from a neural model are treated as System 1 proposals, and a logical, deliberative System 2 filters these proposals for consistency and soundness (see Figure 1). We further take inspiration from the fact that humans often do not need explicit supervision to reason about new problems or domains (e.g., see human evaluation task in Section 4.2) and require that the System 2 module not need additional problem-specific training, especially on example contradictions or commonsense violations. People can handle novelty by reconfiguring, rather than retraining, their internal models (Lake et al., 2017), and we strive to build machine systems capable of the same. We show how a lightweight, easy-to-implement System 2 model can help improve coherence and consistency by adding a small amount of symbolic reasoning.

We tackle two kinds of domains: text generation and instruction following. In both cases, we construct generative models over sequences by using a neural generation model to propose candidate generations and a symbolic world model that can accept or reject the generations and resample proposals if necessary. We first illustrate the approach by generating short stories based on the bAbI dataset (Weston et al., 2015); this pedagogical, synthetic example illustrates how basic commonsense knowledge of objects, agents, and places can inform a text generation model. We then test our approach on rich, natural language vignettes based on CLUTRR (Sinha et al., 2019), focusing on ensuring consistency of family and interpersonal relationships. In both text generation domains, we interface between the explicit logical knowledge/reasoning of System 2 and generations of System 1 using a few-shot learning approach with state-of-the-art neural language models (GPT-3), which requires no additional training or fine-tuning. Even using off-the-shelf transformers and symbolic solvers, our dual-system model improves the consistency and coherence of text generations as measured by human judges. We test our approach also on instruction following, showing how goal-prediction models and execution models can easily be combined to achieve improved performance in low-data regimes. We show improvements over previous work in the gSCAN grounded compositional challenge (Ruis et al., 2020); a dual-system model requires much less data to train than previous models, and achieves higher accuracy and stronger generalization. Overall, our findings indicate that neuro-symbolic, dual process models are a promising means of addressing longstanding problems of robustness and consistency in neural sequence models.

2 Related Work

Our approach incorporates **semantic parsing** (Liang, 2016) as a component of a generative process, where neural generation is used in conjunction with parsing techniques. In our text generation experiments, we employ GPT-3 to perform few-shot semantic parsing without fine-tuning. Related work includes few or zero-shot semantic parsing using pre-training techniques and paraphrasing (Su & Yan, 2017; Herzig & Berant, 2020). It also includes semantic parsing systems trained either without supervision (Liang et al., 2017; Mou et al., 2017; Muhlgay et al., 2019), or with synthetic language data (Marzoev et al., 2020; Xu et al., 2020b).

One popular technique for improving neural generations is **generate-and-rerank**, wherein one model generates proposals and another reranks them. This broad approach has been used in image generation (Ramesh et al., 2021), text generation (Holtzman et al., 2018; Shen et al., 2019; Deng et al., 2020), dialogue systems (for control, coherence and safety (Welleck et al., 2018; Smith et al., 2020; Nie et al., 2020; Xu et al., 2020a)), and instruction following (Kurita & Cho, 2020). Reranking is generally used to improve outputs with respect to relatively broad, holistic criteria. Here, our goal is to make generation robust to particular types of logical errors by pruning with respect to explicit symbolic constraints. Our approach can thus be considered closely related to techniques which employ explicit search to find generations satisfying particular logical constraints. Similar methods, such as guess-and-check or beam search pruning, have had success in neural program synthesis (Devlin et al., 2017; Nye et al., 2020).

Recent work in NLP has used **template-based planning**, in which a model generates text by first generating a plan or skeleton, and filling in the missing words to produce naturalistic text (Xu et al., 2018; Hua & Wang, 2020). To generate stories, Martin et al. (2018) parses previous sentences into events and does planning in event space. Our work extends previous entity/relation/event planning in that the world model is not used for planning, but rather for post-checking candidate generations. Structured parsing of this type is also related to dialog tracking techniques such as slot-filling (Pieraccini et al., 1992). In our work, fully compositional logical facts are extracted from utterances. It is therefore more closely related to systems which extract programs from dialogue, such as Andreas et al. (2020).

Recent work has also studied incorporating symbolic constraints into a neural decoding strategy in the context of natural language. Miao et al. (2019) introduce an MCMC-based inference-time **propose-and-reject** strategy for satisfying constraints. They test on constraints such as paraphrase and grammatical error correction. Lu et al. (2020) introduces "NeuroLogic decoding," which uses logical constraints on neural language models to produce generations which contain (or do not contain) required (or forbidden) keywords. In these works, the constraints are *lexical* or based on word/sentence similarity (and provided in the problem setup for Lu et al. (2020)), whereas we study *logical* constraints on the world state decoded directly from observations or generations at test time.

3 Integrating System 1 and System 2

We introduce our dual-system approach using examples from the bAbI domain (Weston et al., 2015), which we also use to perform diagnostic experiments. Consider generating a simple story involving people, places and objects, such as (from Figure 1):

Daniel went to the garden. Mary traveled to the office. Daniel grabbed the apple.

A model tasked with generating such stories must juggle several simultaneous demands: staying on topic and maintaining consistency of style and other textural elements (for which people rely on System 1), as well as maintaining consistency with previous statements and commonsense knowledge (for which people rely on both systems). Consider continuing the story with one of the following:

(a) *Daniel went to the patio.* (b) *Mary dropped the apple there.*

Sentence (a) is reasonable; sentence (b) is not because it is Daniel, not Mary, who has the apple. During generation, how might a model distinguish between these candidates? Perhaps a well-trained neural language model could track constraints of these sorts. Neural language models to date, however, often violate these types of commonsense, hard constraints without a large high-quality corpus or explicit training on detecting violations of commonsense (Sinha et al., 2019).

We address this problem by decomposing text generation into two parts: candidate generation facilitated by deep neural networks and a logical pruning process implemented via a separate symbolic module. Consider again the example above. To ensure consistency, our model would extract from the text the features of the world that are subject to the hard, logical constraints, such as the location of objects and who is holding them. These constraints can then be checked against an explicit representation of current state of the world. For sentences (a) and (b), the system would extract and `go(Daniel, patio)` and `drop(Mary, apple)`, respectively. A **minimal world model** would track the state of the apple, such that it maintains `apple.holder = Daniel` (or equivalently, `Daniel.inventory = [apple]`). When such a model is given a parse of a candidate generation, `drop(Mary, apple)`, the mismatch between the current state and the proposed change would cause a violation, and the candidate generation will be rejected.

The main steps of our general approach are illustrated in Figure 1: generate proposals from a **System 1 proposal model**, extract facts with a **fact extraction model**, and filter proposed generations by ensuring that they satisfy the constraints given by the extracted facts and the **minimal world model**.

System 1: Generation. We use neural sequence models to produce System 1 generations. In text generation domains, we use a large, pre-trained model that can be fine-tuned or conditioned via a short prompt to generate relevant text. Text sampled from the System 1 model will be treated as candidate utterances, which will be parsed and filtered by System 2 (described below). For the bAbI examples, we use GPT-3 as our **System 1 proposal model** through few-shot prompting with 10 example bAbI stories as context, generating a new story one candidate sentence at a time.

System 2: Fact extraction. A fact extractor, or parser, is used to mediate between the System 1 candidate proposals and the minimal world model within System 2. In our text generation domains, we use a pre-trained GPT-3 model without fine-tuning to perform parsing.

For bAbI, our prompt consist of an initial descriptive sentence *"Please parse the following statements into commands. The available commands are pickup, drop, and go."* and a small set (< 10) of representative semantic parsing examples (input = sentences; output = correct parses, such as `go(Bob, roof)`). The parse of each utterance is produced via few-shot prompting (Brown et al., 2020): the utterance is added to the end of the prompt, and the subsequent GPT-3 generation is interpreted as the target parse. We found that this simple parsing technique works well and could easily be applied to other parsing-based tasks, as in Shin et al. (2021). The parsing prompts are reproduced in full in the Appendix. As discussed in Section 5, for the gSCAN instruction following domain, fact extraction is performed with a learned goal location prediction model.

System 2: Minimal world model. We use a lightweight, incomplete description of the state of the world as a world model in each domain, e.g., commonsense information about the people, objects and locations (Figure 1). The goal is not to track and verify all the possible information; instead, we aim for minimalism, capturing just a few commonsense (or application-critical) variables that we want to ensure are correct. The world model facilitates tracking of long-range logical dependencies and logical consequences, especially those which are not readily decodable from surface forms. The world model also lets us integrate rule-based world-knowledge without retraining (and without the need for a large set of labeled examples).

For the bAbI examples, the minimal world model keeps track of the people, locations and objects introduced in the story so far (Figure 1). This encodes constraints on possible actions related to human core knowledge competencies (objects, agents, places) present early in human development (Spelke & Kinzler, 2007); specifically, a person or object can only be in one place at a time, an object can only be possessed by a single person at a time, a person cannot "go" to a room they are already in, and a person cannot pick up an object if it is in a different room. See the Appendix for details.

Search. At generation time, the interaction between System 1 generation and System 2 parsing yields a neuro-symbolic, guess-and-check search strategy. In a text generation scenario, where text is sampled from the model, our dual-system model improves upon a naive, neural-only sampling method by using the System 2 model to reject candidate utterances which are incompatible with the current state. When a candidate is rejected, a new candidate utterance is sampled from the System 1 model, which is again checked by System 2. This process repeats until a candidate utterance is accepted by System 2 (i.e., the utterance is compatible with the world state). This procedure allows the model to effectively search the space of candidate utterances, guided by the logical constraints from the minimal world model. In this work, we use straightforward probabilistic sampling to illustrate that the approach works with even a very simple search mechanism. We imagine that the search procedure could be further optimized by applying, for example, beam search or stochastic beam sampling.

GPT-3 only:
John went to the bedroom.
John picked up the apple there.
Mary took the apple there.
Mary travelled to the office.
Daniel went back to the garden.
Mary went to the bedroom.
John went to the bedroom.
Sandra went to the bedroom.
Sandra travelled to the office.
Mary went back to the office.
Where is the apple? A: office

GPT-3 + world model:
John went to the bedroom.
John picked up the apple there.
Mary travelled to the office.
Daniel went back to the garden.
Mary went to the bedroom.
Sandra went to the bedroom.
Sandra travelled to the office.
Mary went back to the office.
Where is the apple? A: bedroom

Figure 2: Example bAbI stories generated by GPT-3 only (left) and our dual-system model (right). Logically inconsistent lines are written in red text, and are removed from the story-so-far at generation time.

Diagnostic bAbI experiments. We use Task #2 from bAbI as a diagnostic test for our neuro-symbolic dual-system model. As shown above, this task consists of synthetically-generated short stories involving people, places and objects, and questions concerning the locations of objects in these stories. We investigate performance on both question answering (QA) tasks and story generation. For the QA tasks, we parse each sentence in the story to encode each fact into the world model and parse the final question to query the world model, returning the answer given by the world model. We compare with two alternative models (Table 3 in the Appendix): GPT-3 by itself and a dual-system baseline that uses a neural Natural Language Inference (NLI) model as its System 2. The NLI-based dual-system model generates 10 candidates from GPT-3 and selects the candidate with the highest predicted probability of entailment under the NLI model given the context. We use the RoBERTa MNLI model as our off-the-shelf neural NLI model (Liu et al., 2019), which operates as a System 2 that does not use additional problem-specific data or fine-tuning.[1] On 200 held-out tasks, our GPT-3-based "fact extractor" achieves 100% QA accuracy, far exceeding the performance of GPT-3 alone (29.0%) or GPT-3 generation with neural NLI scoring (32.5%; also see Table 3 in the Appendix). These results show that GPT-3 can be made to answer questions successfully when used for parsing with a world model, even when GPT-3 alone does not achieve high QA accuracy.

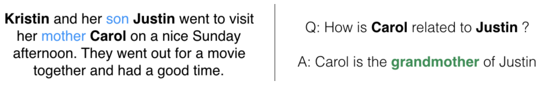
To test story generation, we use our GPT-3-based System 1 proposal model (few-shot prompted on 10 example stories) to sample a new bAbI story, line-by-line. If a generated utterance is inconsistent with the current state as indicated by the System 2 world model, a new utterance is sampled from System 1 (repeating until a consistent utterance is sampled). Figure 2 shows how the dual-system approach generates stories that mimic the statistical structure of bAbI stories, while remaining logically sound In contrast, GPT-3 alone was not able to maintain logical coherence. In a set of 50 generated stories, all stories required at least one sentence to be resampled to maintain coherence, and over half of the generated sentences (53.1%) were rejected by our System 2 model to maintain logical consistency. These results demonstrate that equipping GPT-3 with a minimal world model produces logically coherent stories that mimic the textural structure of the bAbI domain. In the next section, we apply this approach to mimicking human-generated short stories in natural language.

4 Coherent Language Generation - CLUTRR

We apply our dual-system approach to a dataset of natural language using the CLUTRR dataset. CLUTRR contains human-written stories about people and their family relationships (see example in Figure 3). As with bAbI, CLUTRR was originally designed as a Question Answering challenge; instead, we use it to evaluate coherent language generation by querying models to generate complete CLUTRR-style stories or to complete partially-generated stories. Our particular aim is to produce stories with coherent and logically consistent family relationships. As above, our language generation setup consists of pre-trained language models acting as our System 1 proposer, a minimal world model as System 2, and a neural semantic parser (implemented via few-shot GPT-3 prediction) as a bridge between the two systems. We use human judgments to assess whether our neuro-symbolic, dual-system model produces more consistent and coherent stories relative to a baseline.

4.1 Model specification

As our System 1 proposal model, we used pre-trained neural models to produce candidate generations one sentence at a time. We experimented with GPT-3 as our System 1 model (which we



Kristin and her son **Justin** went to visit her mother **Carol** on a nice Sunday afternoon. They went out for a movie together and had a good time.

Q: How is **Carol** related to **Justin** ?

A: Carol is the **grandmother** of Justin

Figure 3: Sample story from the CLUTRR dataset. Each story consists of a sequence of human-generated sentences concerning family relationships. Adapted from Sinha et al. (2019).

[1] Previous work has used domain-specific entailment/contradiction data to train reranking models (Welleck et al., 2018), however, this requires collecting a dataset of domain-specific entailment and contradiction data.

Table 1: Statistics from CLUTRR story generation. We report the percentage of generations (on both a per-line and per-story basis) for which the System 2 world model did not detect an error. The dual-system model is able to detect many inconsistencies in the neural single-system generations, and most can be corrected by re-sampling new candidates (up to a limit of ten).

	% w/out error detected (per line)		% w/out error detected (per story)	
	single-system	dual-system	single-system	dual-system
	(neural gen. only)	(neural gen.+world model)	(neural gen. only)	(neural gen.+world model)
prompt from dataset	82.8	97.1	60	96.1
prompt from model	71.9	96.3	36.4	93.5

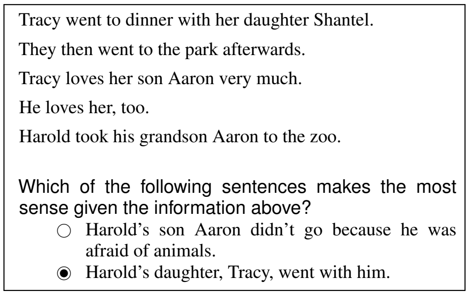
used above for bAbI), but found generations too unreliable, often outputting the empty string. Instead, we used a BART model (Lewis et al., 2019) that was fine-tuned on the CLUTRR training corpus. This model also gives us an opportunity to compare against a best-case neural "single-system" baseline, specifically fine-tuned on story data. To maintain a state of family relations, we use a constraint solver in our "System 2" to encode family relationships (e.g., `child(x, y)`, `spouse(x, z)`) and check that the candidate utterances do not contradict the previous statements (e.g., a person cannot be their own child or married to their sibling). We implemented the world model as a set of logical relations and constraints using the Z3 solver (De Moura & Bjørner, 2008). For instance, we require that the parent of `x` cannot also be the uncle of `x`: For all `x, y`, `uncle(x, y)` $\Rightarrow \neg$`child(y, x)`. To check a candidate utterance, we query the solver to determine if the set of constraints is satisfiable or if there is a contradiction. The full set of constraints and other details can be found in the Appendix. We again used GPT-3 as our semantic parser, extracting parses for each candidate utterance via few-shot learning. This parsing approach worked well, even for the natural language in this domain. We observed that parsing with GPT-3 was more successful when the target parse was naturalistic, i.e., *"Bob is Joe's father."* rather than *"father(Bob, Joe)"*. The parsing prompt is reproduced in full in the Appendix.

4.2 Human judgments

We test our dual-system neural generation + world model method in its ability to generate stories that are deemed by naive human participants to be more naturalistic and coherent than those generated from the baseline models. Specifically, we asked participants to select which of two continuations made the most sense to them, where one continuation was generated from the neural model alone (single-system) and the other from a dual-system model (either the world model System 2 or the neural NLI System 2).

Participants. Participants (N = 101) were recruited on the crowd-sourcing platform Prolific and compensated \$2 for the task (∼15 minutes, so roughly \$8/hour). Participants gave informed consent, and the study was approved by MIT's IRB. 21 participants were excluded for failing an instruction quiz, incorrectly answering more than one of five filler questions, or finishing the task too quickly. The data we collected contains no personally identifiable information or offensive content.

Procedure. Participants began the experiment by reading a set of instructions and answering comprehension questions. On each main trial, participants were shown a prompt consisting of several sentences



Tracy went to dinner with her daughter Shantel.

They then went to the park afterwards.

Tracy loves her son Aaron very much.

He loves her, too.

Harold took his grandson Aaron to the zoo.

Which of the following sentences makes the most sense given the information above?

○ Harold's son Aaron didn't go because he was afraid of animals.

● Harold's daughter, Tracy, went with him.

Figure 4: Example trial from CLUTRR human judgement experiment. Participants were instructed to select which of two options makes the most sense given the prompt. One option was generated by the System 1 model only ("single-system"), while the other was generated by the dual-system model.

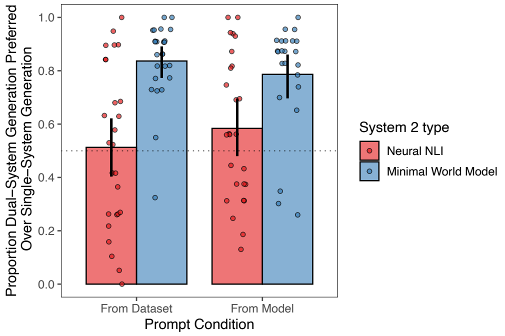


Figure 5: CLUTRR human judgment experiment results. Bars denote proportions of dual-system generations selected as making more sense over single-system generations, in each of four conditions. Error-bars denote bootstrapped 95% confidence intervals of the item means. The points denote means for each individual item in the experiment and are jittered horizontally for clarity.

and were asked to choose which of two possible continuations made the most sense (an example trial is shown in Figure 4). Participants were instructed that if a name appeared multiple times within a trial, then it referred to the same person, whereas if a name appeared across trials, then it was not referring to the same person. For each trial, one continuation option was generated by the neural only single-system baseline, while the other was a dual-system generation. We selected generations from the neural only baseline that were rejected by the System 2 model in order to maximize the differences between the models' generations; thus, human judgments pertain to generations that the models disagreed on. Each participant performed between 20 and 26 trials.

Materials. Participants were randomly assigned to one of four between-participant conditions, which varied according to the kind of prompt and the kind of dual-system model. The prompt was either generated from the model (up to the point of disagreement between System 1 and System 2 models; "Prompts from model" condition) or taken completely from the length 4 CLUTRR systematic generalization test dataset ("Prompts from dataset" condition). To generate prompts for the "from model" condition, we took the first sentence of each story from the CLUTRR test dataset and generated subsequent prompt sentences from the dual-system model; sentences were generated until the two systems disagreed (i.e., System 1 generated a sentence that System 2 rejected), at which point the "rejected sentence" served as the neural only (single-system) baseline generation and the first resampled sentence that System 2 accepted served as the dual-system generation. Prompts were sampled to a maximum length of four sentences. The dual-system model shown to participants used a System 2 based on either our constraint-based "world model" or the neural NLI baseline.

Table 1 catalogs critical statistics from the stimulus generation process. We generated vignettes from the System 1 model and report the percentage of System 1 generations which are deemed correct by the System 2 model.[2] We also report the percentage of generations corrected by the System 2 model (i.e., if System 1 made an error, could System 2 fix it within 10 attempts?). We report these statistics on both a per-story and per-line basis. According to System 2, the System 1 generation model makes a lot of errors (only 36.4% of stories and 71.9% of lines were error-free, in the "from model" condition). In most instances, re-sampling new generations yields stories that, according to System 2, no longer contain logical errors within a budget of 10 samples (93.5% of stories and 96.3% of lines were error-free, respectively).

Results. The human evaluation indicates that System 2 is indeed correcting genuine errors in the stories. As summarized in Figure 5, participants strongly preferred the dual-system neural generation + world model continuations in comparison to the neural only single-system continuations (proportion preferring dual-system = 0.84; bootstrapped 95% confidence interval [0.77, 0.89] and 0.79 [0.77, 0.89] for the "from dataset" and "from model" prompt conditions, respectively). The dual-system approach, however, did not improve generation quality when the System 2 was based on an off-the-shelf neural NLI model (Proportion preferring dual-system = 0.51; [0.40, 0.64] for "from dataset"; 0.58 [0.48, 0.68] for "from model"). Thus, when using a minimal world model, the dual-system approach dramatically improves logical consistency without any need for additional training or fine-tuning. People clearly prefer neuro-symbolic generations from the dual-system model over purely neural generations from a single-system model.

5 Grounded Instruction Following

The dual-system approach offers a general-purpose means of improving upon generative, neural sequence models by incorporating logical constraints. To highlight its generality, we examine how the dual-system perspective can be deployed in a very different domain: grounded instruction following. In Heinze-Deml & Bouchacourt (2020), a learned target location predictor was used to increase the accuracy of a neural action sequence generation model. Here, we show how to increase performance further by enforcing consistency between the target location predictor and the action sequence generator in our dual-system framework.

We use the gSCAN benchmark (Ruis et al., 2020), a recently proposed grounded instruction following dataset designed to measure compositional generalization in neural systems. Given an initial gridworld state and an instruction, e.g., "walk to the big square," an agent must predict the sequence of low-level actions which achieve the goal, e.g., "`TURN LEFT, WALK, TURN LEFT, WALK`" (See Figure 6). The dataset contains several test splits, each testing different aspects of compositional generalization.

[2] For all System 1 generations, we used model temperature of 1.0. For the neural NLI baseline, we used 0.9 probability of contradiction as the cutoff for rejection. Our dual-system model uses a sampling budget of 10 System 1 samples per sentence. Contradictions remaining after 10 samples are considered dual-system errors.

Table 2: Accuracy on gSCAN splits. Models were trained on 5000 examples (only 2.5% of the gSCAN training data). See Appendix Table 4 for additional results.)

Test split:	single-system[3]	dual-system
dev	71.7	83.3
random	57.2	74.7
yellow squares	68.1	81.3
red squares	64.9	78.1
novel direction	0.0	0.01
relativity	41.0	53.6
class inference	68.1	76.2
adverb (k=1)	0.0	0.0
adverb to verb	20.8	21.8

[3] From Heinze-Deml & Bouchacourt (2020)

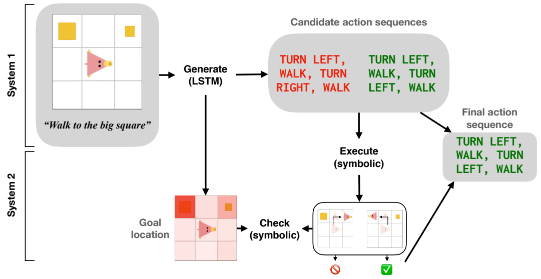

System 1
"Walk to the big square"
Generate (LSTM)
Candidate action sequences
TURN LEFT, WALK, TURN RIGHT, WALK
TURN LEFT, WALK, TURN LEFT, WALK
Final action sequence
TURN LEFT, WALK, TURN LEFT, WALK
Execute (symbolic)
System 2
Goal location
Check (symbolic)

Figure 6: Schematic of our dual-system approach to gSCAN. We train a neural sequence model to predict both a distribution over action sequences, and a distribution over target locations. At test time, we decode candidate action sequences from the model, execute them on the gridworld, and only accept a sequence that brings the agent to the predicted target location (shown in green).

Our model builds on Heinze-Deml & Bouchacourt (2020) by using an LSTM to predict the correct action sequence and target location. Given a command c and an initial gridworld state s, the neural network defines two distributions: a distribution over action sequences $q_a(a|c, s)$ and a distribution over target grid locations $q_{loc}(l|c, s)$. Heinze-Deml & Bouchacourt (2020) showed that when these distributions share parameters, using location prediction as an auxiliary loss improves the accuracy of the action sequence prediction model. We can further exploit these two models by noticing that when a predicted action sequence is not consistent with a predicted target location, then either the action sequence or the target location must be incorrect. Since the target location is much simpler to predict, and thus much more likely to be correctly predicted, if a predicted action sequence is not consistent with the predicted target location, then the action sequence is most likely incorrect. Our dual-system framework can use this property to increase action sequence prediction accuracy. Consider the initial state and command in Figure 6. Our model predicts candidate action sequences, and also predicts that the most likely target location is the grid containing the bigger yellow square (highlighted in red). The model then executes the candidate action sequences, and only accepts a sequence which results in the agent standing in the target location.

In the language of our dual-system approach, we treat the distribution over actions $q_a(a|c, s)$ as our **System 1 proposal model**. The distribution over target locations $q_{loc}(l|c, s)$ serves as a **fact extractor** model, which extract a location constraint l. As a **minimal world model**, we use a deterministic gridworld execution model $T(a, s_0) \to s_f$, which takes a state and action and predicts the resulting state. At test time, we first extract the predicted location as $l = \arg\max_{l'} q_{loc}(l'|c)$. We then search through the possible action sequences from $q_a(\cdot|c)$, conditioned on agreement with l. In our experiments, we use a sample-based search with a maximum budget of 50 samples. We trained models on random subsets of the gSCAN training set of varying sizes: 5000 datapoints, 8000 datapoints, and 20000 datapoints (2.5%, 4% and 10% of the original training set, respectively).

Results. The results show that the System 2 execution model improves performance without the need for any additional training (see Table 2 for results training on 5000 examples). In contrast to the single-system model, the dual-system model allows for sampling many candidate action sequences from the neural network, accepting only consistent sequences. This guess-and-check approach greatly increases the evaluation accuracy, improving upon prior work on gSCAN, particularly in low-data regimes.

6 Limitations

In its current form, our approach is most useful in domains where naturalistic, learned generation is necessary and where a small number of mission-critical logical constraints can be explicitly articulated. Our system will be less useful when constraints are more difficult to articulate (e.g., creative domains such as writing poetry) or when there are many constraints, since the minimal world model must be hand-engineered. Enforcing strict constraints may also pose risks: if the constraints are not only logical but cultural, they may be harmful if misapplied. However, these constraints must be articulated explicitly in a symbolic model, and are thus easier to identify and correct.

The current few-shot parsing technique may also suffer from a limited capacity. For more complex domains, the number of examples required to specify the desired parsing behavior may be too large (i.e., they may not fit in the input window) or too complex for a model to perform parsing accurately. While some tasks may not be suitable, the complexity of the world model need not necessarily increase hand-in-hand with the complexity of the application domain. A dual-system model will be most successful when tracking just a few critical variables (e.g., tracking consistency in family relations, as in our experiments, or tracking scheduling constraints when discussing a team plan).

A promising direction for future work is to incorporate learning into the System 2 world model. Currently, the minimal world knowledge that exists in System 2 can be easily modified, but changes must be made by hand. Improvements would come from automatically learning and updating this structured knowledge, possibly by incorporating neuro-symbolic learning techniques (Ellis et al., 2020; Mao et al., 2019).

Learning could improve our dual-system approach in other ways, e.g., by training a neural module to mimic the actions of a symbolic System 2. The symbolic System 2 judgments could be used as a source of supervision; candidate utterances rejected by the symbolic System 2 model could be used as examples of contradictory sentences, and accepted utterances could be used as examples of non-contradictory statements. This oversight could help train a neural System 2 contradiction-detection model capable of more subtleties than its symbolic counterpart, especially in domains where labeled examples are otherwise unavailable. This approach may also help us understand aspects of human learning, where certain tasks that require slower, logical reasoning can be habitualized over time and tackled by faster, more intuitive reasoning.

Recent work (Li et al., 2021) has shown that large pre-trained neural models learn to approximately represent certain types of structured semantic information. However, it is not yet clear how representational fidelity translates to logical coherence during generative tasks. Our current approach allows us to explicitly fix logical errors in generation, which may ultimately be caused by representational errors. Understanding how we might leverage our approach to improve the representation of structured knowledge within neural models is a promising direction for future work, which could lead to increased generation consistency and coherence.

7 Conclusion

Inspired by dual process theories from cognitive science, we combine the respective strengths of neural and symbolic approaches to build more robust models that can more effectively incorporate domain knowledge. For language generation, we showed that equipping neural generation with a minimal symbolic world model increased language coherence and consistency. For grounded instruction following, we showed that requiring test-time consistency between predicted action sequences and goal locations led to improved performance, especially in low-data regimes. Our neuro-symbolic approach can readily be applied to other domains and types of prior knowledge, as a lightweight way of improving the coherence and consistency of powerful neural sequence models.

This paper just scratches the surface of how structured knowledge can make neural systems more robust; we hope to inspire further work into neuro-symbolic systems which possess the robustness and commonsense necessary for human-level intelligence.

Acknowledgments

We thank Laura Ruis, Jacob Andreas, Yewen (Evan) Pu, Joe O'Connor and Guy Davidson for helpful comments on an earlier version of this manuscript.

References

Andreas, J., Bufe, J., Burkett, D., Chen, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., et al. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

De Moura, L. and Bjørner, N. Z3: An efficient smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337–340. Springer, 2008.

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Marc'Aurelio Ranzato. Residual energy-based models of text generation. In *International Conference on Learning Representations (ICLR)*, pp. 1–18, 2020.

Devlin, J., Uesato, J., Bhupatiraju, S., Singh, R., Mohamed, A.-r., and Kohli, P. Robustfill: Neural program learning under noisy i/o. *ICML*, 2017.

Ellis, K., Wong, C., Nye, M., Sable-Meyer, M., Cary, L., Morales, L., Hewitt, L., Solar-Lezama, A., and Tenenbaum, J. B. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.

Evans, J. S. B. In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences*, 7(10): 454–459, 2003.

Frederick, S. Cognitive reflection and decision making. *Journal of Economic perspectives*, 19(4):25–42, 2005.

Garcez, A. d. and Lamb, L. C. Neurosymbolic ai: The 3rd wave. *arXiv preprint arXiv:2012.05876*, 2020.

Goyal, A. and Bengio, Y. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020.

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

Herzig, J. and Berant, J. Span-based semantic parsing for compositional generalization. *arXiv preprint arXiv:2009.06040*, 2020.

Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., and Choi, Y. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*, 2018.

Hua, X. and Wang, L. Pair: Planning and iterative refinement in pre-trained transformers for long text generation. In *EMNLP*, 2020.

Kahneman, D. Thinking, fast and slow. kindle ed, 2013.

Kurita, S. and Cho, K. Generative language-grounded policy in vision-and-language navigation with bayes' rule. *arXiv preprint arXiv:2009.07783*, 2020.

Lake, B. M. and Murphy, G. L. Word meaning in minds and machines. *arXiv preprint arXiv:2008.01766*, 2020.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:E253, 2017.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Li, B. Z., Nye, M., and Andreas, J. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

Liang, C., Berant, J., Le, Q. V., Forbus, K., and Lao, N. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23–33, Vancouver, Canada, 2017. URL http://aclanthology.coli.uni-saarland.de/pdf/P/P17/P17-1003.pdf.

Liang, P. Learning executable semantic parsers for natural language understanding. *Communications of the ACM*, 59(9):68–76, 2016.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, X., West, P., Zellers, R., Bras, R. L., Bhagavatula, C., and Choi, Y. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*, 2020.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

Martin, L. J., Ammanabrolu, P., Hancock, W., Singh, S., Harrison, B., and Riedl, M. O. Event representations for automated story generation with deep neural nets. In *AAAI*, 2018.

Marzoev, A., Madden, S., Kaashoek, M. F., Cafarella, M., and Andreas, J. Unnatural language processing: Bridging the gap between synthetic and natural language data. *arXiv preprint arXiv:2004.13645*, 2020.

Miao, N., Zhou, H., Mou, L., Yan, R., and Li, L. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *AAAI*, 2019.

Mou, L., Lu, Z., Li, H., and Jin, Z. Coupling distributed and symbolic execution for natural language queries. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2518–2526. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/mou17a.html.

Muhlgay, D., Herzig, J., and Berant, J. Value-based search in execution space for mapping instructions to programs. pp. 1942–1954, 01 2019. doi: 10.18653/v1/N19-1193.

Nie, Y., Williamson, M., Bansal, M., Kiela, D., and Weston, J. I like fish, especially dolphins: Addressing contradictions in dialogue modelling. *arXiv preprint arXiv:2012.13391*, 2020.

Nye, M. I., Solar-Lezama, A., Tenenbaum, J. B., and Lake, B. M. Learning compositional rules via neural program synthesis. *arXiv preprint arXiv:2003.05562*, 2020.

Pieraccini, R., Tzoukermann, E., Gorelov, Z., Gauvain, J.-L., Levin, E., Lee, C.-H., and Wilpon, J. A speech understanding system based on statistical representation of semantics. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pp. 193–196 vol.1, 1992. doi: 10.1109/ICASSP.1992.225939.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., and Lake, B. M. A benchmark for systematic generalization in grounded language understanding. *arXiv preprint arXiv:2003.05161*, 2020.

Shen, S., Fried, D., Andreas, J., and Klein, D. Pragmatically informative text generation. *arXiv preprint arXiv:1904.01301*, 2019.

Shin, R., Lin, C. H., Thomson, S., Chen, C., Roy, S., Platanios, E. A., Pauls, A., Klein, D., Eisner, J., and Van Durme, B. Constrained language models yield few-shot semantic parsers. *arXiv preprint arXiv:2104.08768*, 2021.

Sinha, K., Sodhani, S., Dong, J., Pineau, J., and Hamilton, W. L. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019.

Smith, E. M., Gonzalez-Rico, D., Dinan, E., and Boureau, Y.-L. Controlling style in generated dialogue. *arXiv preprint arXiv:2009.10855*, 2020.

Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.

Su, Y. and Yan, X. Cross-domain semantic parsing via paraphrasing. *arXiv preprint arXiv:1704.05974*, 2017.

Welleck, S., Weston, J., Szlam, A., and Cho, K. Dialogue natural language inference. *arXiv preprint arXiv:1811.00671*, 2018.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Xu, J., Ren, X., Zhang, Y., Zeng, Q., Cai, X., and Sun, X. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4306–4315, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1462. URL https://www.aclweb.org/anthology/D18-1462.

Xu, J., Ju, D., Li, M., Boureau, Y.-L., Weston, J., and Dinan, E. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*, 2020a.

Xu, S., Semnani, S. J., Campagna, G., and Lam, M. S. Autoqa: From databases to qa semantic parsers with only synthetic training data. *arXiv preprint arXiv:2010.04806*, 2020b.

A Additional CRT experiments

Figure 8 shows additional cognitive reflection test (CRT) experiments using GPT-3. For each question type, the original is displayed along with a table containing the responses for several variants. Prompt text is in black, and response text is in color (green for correct, red for incorrect). Since it is unknown whether the CRT tasks are within the GPT-3 training set, we test several conceptually similar variants. All experiments were performed with the “davinci" model using temperature 0 and the newline token set as the stop sequence. For each question, GPT-3 often (but not always) makes the same mistakes as humans, even when the numbers are modified from the original CRT task.

B Experimental Details

For all experiments using GPT-3, we used the largest available “davinci" model. All other models were implemented in PyTorch. All testing and training was performed on one Nvidia GTX 1080 Ti GPU. Language generation experiments generally took less than 4 hours to run, and gSCAN experiments took less than 48 hours.

B.1 bAbI

Table 3: Question answering results on bAbI.

Model	Acc.
GPT-3 generation only	29.0%
GPT-3 generation, neural NLI scoring	32.5%
parsing (via GPT-3) + Sys 2 world model	100%

The full text of the parsing prompts for the bAbI domain can be found in Figure 9. QA results are shown in Table 3.

Minimal world model. The minimal world model for bAbI is implemented via simple Python code and performs three functions:

1. Tracks the people, objects and locations which have been mentioned so far.
2. Modifies the world state changes as a result of parsed actions.
3. Checks if the candidate action violates the current world state, as defined by (1) and (2).

Tracking the people, objects and locations is performed by maintaining a lookup table which maps the string representing a name (e.g., *‘football’*) to the Python object that represents the corresponding person, object or location. The Python objects inherit from one of three small classes, `Person`, `Location`, or `Obj`. When a new person, object or location is referenced, a new corresponding Python object is initialized and added to the lookup table. The logic for possible actions (in our experiments, `pickup`, `go`, and `drop`) are implemented to carry out the intended action. For instance, `pickup(person, obj)` adds `obj` to the inventory of `person`. Likewise, `go(person, location)` changes the location of `person` and `person`’s inventory to `location`. Each action checks whether the current world state satisfies necessary preconditions for the action. If the current world state violates the action preconditions, an error is thrown, and the candidate action is rejected. For example, if the location of `obj` (`obj.location`) has been specified and is not the same as the location of `person`, then `pickup(person, obj)` will fail.

The full world model used for bAbI (including code for interpreting the output of the GPT-3 fact extractor) consists of fewer than 200 lines of code, and can be found in `worldModel.py`.

B.2 CLUTRR

For our CLUTRR experiments, we used a BART-base model (Lewis et al., 2019), which was fine-tuned on the “systematic generalization" training corpus with story lengths of 2-4 sentences from the CLUTRR dataset (folder `data_db9b8f04` from the dataset found at `github.com/facebookresearch/clutrr`). BART models were retrieved from Hugging Face and fine-tuned with the Hugging Face default AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of 1e-5 and dropout of 0.1. The full text of the parsing prompt for the CLUTRR domain is shown in Figure 10. Statistics for the CLUTRR story generation for both the symbolic world model and neural NLI System 2 models are shown in Table 5. Additional trials from the human judgement experiment are shown in Figure 11.

Table 4: Results for each gSCAN split, showing exact match accuracy. The dual-system model outperforms baselines in nearly all test splits, and is especially beneficial in the low-data regime.

	5000 examples		8000 examples		20000 examples	
test split:	single-system[4]	dual-system	single-system	dual-system	single-system	dual-system
dev	71.7	83.3	81.0	90.7	92.6	96.9
random	57.2	74.7	69.1	84.6	88.6	95.2
yellow squares	68.1	81.3	77.2	89.3	89.1	94.2
red squares	64.9	78.1	76.3	88.1	87.8	93.2
novel direction	0.0	0.01	0.0	0.1	0.0	0.02
relativity	41.0	53.6	40.0	50.1	62.5	68.9
class inference	68.1	76.2	78.1	87.6	89.4	96.6
adverb (k=1)	0.0	0.0	0.0	0.0	0.0	0.0
adverb to verb	20.8	21.8	20.2	20.6	19.9	21.4

[4] From Heinze-Deml & Bouchacourt (2020)

The family relation constraints used for the world model can be found in Figure 7. The code implemented these constraints using the Z3 solver can be found in `clutrrZ3.py`. All constraints are gender neutral, and read left to right, e.g., `child`$(x, y) \rightarrow$ *the child of x is y*. We follow the notation used in Sinha et al. (2019), and use the term `grand` for *grandchild*, `un` for *aunt/uncle*, etc., and use the prefix `inv_` for inverses of already-defined terms.

We accept the following terms as outputs of the GPT-3-based parsing system, mapping them to the correct constraint: *'spouse', 'husband', 'wife', 'parent', 'grandchild', 'granddaughter', 'grandson', 'grandparent', 'grandmother', 'grandfather', 'father', 'mother', 'uncle', 'aunt', 'nephew', 'niece', 'sister', 'brother', 'daughter', 'son', 'daughter in law', 'son in law', 'mother in law', 'father in law', 'mother-in-law', 'father-in-law', 'daughter-in-law', 'son-in-law.'* All other terms output from the parsing system (e.g., 'Mark is Mary's *friend*') are ignored and do not lead to the addition of new constraints to the current world state.

B.3 gSCAN

Table 4 shows the accuracy results for gSCAN on training sets of size 5000, 8000, and 20000.

Architecture Our model architecture is identical to the “Both" attentional variant from Heinze-Deml & Bouchacourt (2020). In this model, a BiLSTM encoder encodes the instruction sequence and a CNN encoder encodes the initial gridworld state. To predict a target location, attention-weighted gridworld state encodings (which also attend over the instruction sequence encoding) are passed to a linear classification layer, which predicts softmax scores for each gridworld position. To predict an action sequence, an LSTM decodes the action sequence based on attention weighted instruction encodings and attention weighted gridworld encodings. The gridworld encoding vectors are additionally weighted by the softmax scores from the target location prediction layer before being fed to the LSTM decoder attention mechanism. We use the same hyperparameters as Heinze-Deml & Bouchacourt (2020), including a CNN dropout rate of 0.1, a dropout rate of 0.3, an auxiliary task weight of 0.3, and LSTM sizes of 100. See Heinze-Deml & Bouchacourt (2020) for more details.

Test-time search. At test time, the neural only single-system baseline from Heinze-Deml & Bouchacourt (2020) performs greedy decoding. To directly compare to the single-system model, the dual-system model performed greedy decoding to produce the first candidate action sequence. If this candidate action sequence failed the consistency check, the model proceeded with a sample-based search, as described above, with a sampling budget of 50 samples.

Consistency check. The target location prediction in Heinze-Deml & Bouchacourt (2020) predicts the initial location of the target object. However, to successfully complete some of the actions in the gSCAN domain, such as “push" and “pull", the agent must move the target object to a different grid. Therefore, the final grid of the agent is not always the same as the target location. To account for this, the consistency check required only that the agent *passes through* the target location and does not move outside the bounds of the gridworld. This is a strictly less stringent constraint than requiring that the agent's final location matches the target location; nevertheless, we see that this constraint is sufficient to achieve significant accuracy gains.

Relations:
$\forall x, y, z.\texttt{child}(x, z) \wedge \texttt{child}(z, y) \implies \texttt{grand}(x, y)$
$\forall x, y, z.\texttt{grand}(x, z) \wedge \texttt{sibling}(z, y) \implies \texttt{grand}(x, y)$
$\forall x, y, z.\texttt{inv\_child}(x, z) \wedge \texttt{inv\_child}(z, y) \implies \texttt{inv\_grand}(x, y)$
$\forall x, y, z.\texttt{sibling}(x, z) \wedge \texttt{inv\_grand}(z, y) \implies \texttt{inv\_grand}(x, y)$
$\forall x, y, z.\texttt{child}(x, z) \wedge \texttt{sibling}(z, y) \implies \texttt{child}(x, y)$
$\forall x, y, z.\texttt{SO}(x, z) \wedge \texttt{child}(z, y) \implies \texttt{child}(x, y)$
$\forall x, y, z.\texttt{sibling}(x, z) \wedge \texttt{inv\_child}(z, y) \implies \texttt{inv\_child}(x, y)$
$\forall x, y, z.\texttt{child}(x, z) \wedge \texttt{inv\_grand}(z, y) \implies \texttt{inv\_child}(x, y)$
$\forall x, y, z.x \neq y \wedge \texttt{inv\_child}(x, z) \wedge \texttt{child}(z, y) \implies \texttt{sibling}(x, y)$
$\forall x, y, z.\texttt{child}(x, z) \wedge \texttt{SO}(z, y) \implies \texttt{in\_law}(x, y)$
$\forall x, y, z.\texttt{SO}(x, z) \wedge \texttt{inv\_child}(z, y) \implies \texttt{inv\_in\_law}(x, y)$

$\forall x, y, z.\texttt{sibling}(x, z) \wedge \texttt{child}(z, y) \implies \texttt{inv\_un}(x, y)$
$\forall x, y, z.\texttt{inv\_child}(x, z) \wedge \texttt{sibling}(z, y) \implies \texttt{un}(x, y)$

Inverses:
$\forall x, y.\texttt{child}(x, y) \iff \texttt{inv\_child}(y, x)$
$\forall x, y.\texttt{inv\_in\_law}(x, y) \iff \texttt{in\_law}(y, x)$
$\forall x, y.\texttt{inv\_grand}(x, y) \iff \texttt{grand}(y, x)$
$\forall x, y.\texttt{inv\_un}(x, y) \iff \texttt{un}(y, x)$

Symmetric rules:
$\forall x, y.\texttt{sibling}(x, y) \iff \texttt{sibling}(y, x)$
$\forall x, y.\texttt{SO}(x, y) \iff \texttt{SO}(y, x)$

Definition of an ancestor:
$\forall x, y.\texttt{child}(x, y) \implies \texttt{ancestor}(y, x)$
$\forall x, y.\texttt{grand}(x, y) \implies \texttt{ancestor}(y, x)$

Sibling is transitive:
$\forall x, y, z.x \neq z \wedge \texttt{sibling}(x, y) \wedge \texttt{sibling}(y, z) \implies \texttt{sibling}(x, z)$

You can't be your own ancestor:
$\forall x, y.\texttt{ancestor}(x, y) \implies \neg\texttt{ancestor}(y, x)$ Ancestor transitivity:
$\forall x, y, z.\texttt{ancestor}(x, y) \wedge \texttt{ancestor}(y, z) \implies \texttt{ancestor}(x, z)$
You can't be your own sibling:
$\forall x.\neg\texttt{sibling}(x, x)$
Inverse of aunt/uncle:
$\forall x, y.\texttt{inv\_un}(x, y) \implies (\exists z.\texttt{sibling}(x, z) \wedge \texttt{child}(z, y))$
Inverse of sibling:
$\forall x, y.\texttt{sibling}(x, y) \implies (\exists z.\texttt{child}(z, x) \wedge \texttt{child}(z, y))$
Parents can't be aunts/uncles:
$\forall x, y.\texttt{un}(x, y) \implies \neg\texttt{inv\_child}(x, y) \, \forall x, y.\texttt{inv\_child}(x, y) \implies \neg\texttt{un}(x, y)$

Figure 7: Family constraints for CLUTRR world model. Based on constraints used in Sinha et al. (2019). All constraints are gender neutral, and read as left to right, e.g., $\texttt{child}(x, y) \rightarrow$ *the child of x is y.*

Table 5: Statistics from CLUTRR story generation.

	neural NLI System 2				minimal world model System 2			
	% w/out error detected (per line)		per story		per line		per story	
	single-system	dual-system	single-system	dual-system	single-system	dual-system	single-system	dual-system
prompt from model	73.3	98.4	40.2	94.8	71.9	96.3	36.4	93.5
prompt from dataset	82.9	100	57.1	100	82.8	97.1	60	96.1

A ball and a bat cost **$1.10.**
The bat costs one dollar more than the ball.
How much does the ball cost?
Answer: 10 cents

Total cost	**Response**
$1.10	10 cents
$1.20	20 cents
$1.30	$0.30
$1.70	$0.70

If it takes **5** machines **5** minutes to make **5** widgets, how long would it take 100 machines to make 100 widgets?
Answer: 5 minutes.

Value	**Response**
5	5 minutes
6	6 minutes
7	7 minutes
8	8 minutes
9	9 minutes
10	10 minutes
11	100 minutes
12	100 minutes
13	100 minutes

In a lake, there is a patch of lily pads.
Every day, the patch doubles in size.
If it takes **48** days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?
Answer: 24 days

# of days	**Response**
8	(no output)
20	10 days
24	12 days
32	(no output)
25	(no output)
36	(no output)
48	24 days
100	50 days

Figure 8: GPT-3 responses to CRT problems and variants. Correct answers shown in green, and incorrect answers shown in red. “(no output)" indicates that the model produced the newline token (set as the stop sequence) and did not produce an output.

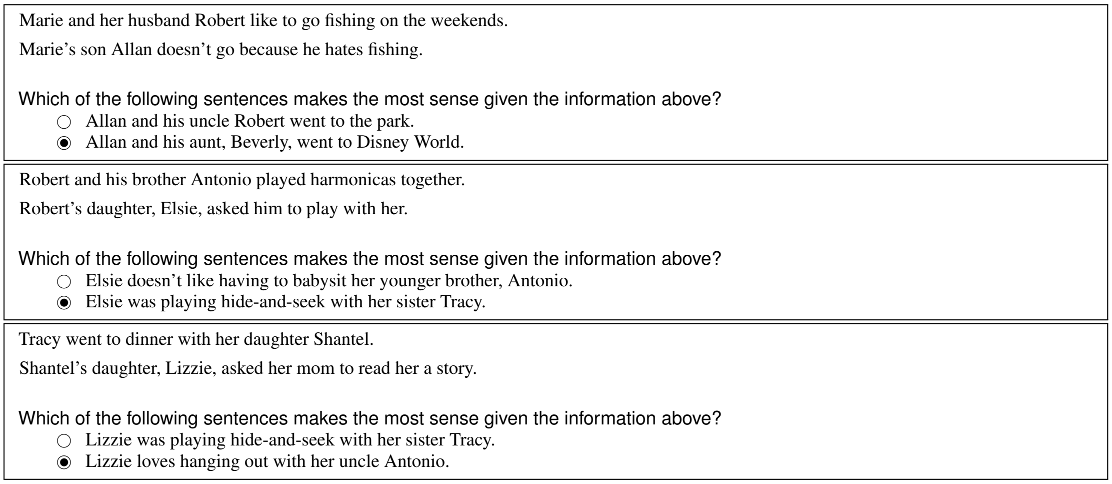
```
Please parse the following statements into commands. The available commands are pickup, drop, and go.
Sentence: Max journeyed to the bathroom.
Semantic parse: go(Max, bathroom)

Sentence: Mary grabbed the football there.
Semantic parse: pickup(Mary, football)

Sentence: Bob picked up the apple.
Semantic parse: pickup(Bob, apple)

Sentence: Susan dropped the milk.
Semantic parse: drop(Susan, milk)

Sentence: Bob got the football there.
Semantic parse: pickup(Bob, football)

Sentence: Max left the cup.
Semantic parse: drop(Max, cup)

Sentence: Kevin put down the pie there.
Semantic parse: drop(Kevin, pie)

Sentence: John took the football there.
Semantic parse: pickup(John, football)

Sentence:
```

```
Please parse the following questions into queries using queryObjLoc:
Question: Where is the toothbrush?
Semantic parse: queryObjLoc(toothbrush)

Question: Where is the milk?
Semantic parse: queryObjLoc(milk)

Question: Where is the apple?
Semantic parse: queryObjLoc(apple)

Question: Where is the football?
Semantic parse: queryObjLoc(football)

Question:
```

Figure 9: Semantic parsing prompts for bAbI domain. Top: Statement semantic parsing prompt. Bottom: Question semantic parsing prompt.

```
The following sentences contain people and their family relationships. Please parse each sentence into family relationships.
The available relationships are sibling, parent, child, grandchild, uncle, spouse.
If a sentence has no relationship, say "None".

Sentence: Michael’s sister, Mary, was crying, so he told her a joke.
Semantic parse: Mary is Michael’s sister.

Sentence: Joshua’s son, Clarence, loves trains.
Semantic parse: Clarence is Joshua’s child.

Sentence: David is very lucky to have a husband who adores her and treats her like a queen.
Semantic parse: None

Sentence: Lillian is married to Thomas and when she was 24, the couple welcomed April into the world.
Semantic parse: Thomas is Lillian’s spouse. April is Lillian’s child.

Sentence: Bobby does n’t like his grandfather James.
Semantic parse: Bobby is James’s grandchild.

Sentence: He loved spending time with her, and she loved it too.
Semantic Parse: None

Sentence: Jerry asked his father George if he could borrow some money.
Semantic parse: Jerry is George’s child

Sentence: Robert and his brother Louis watch Robert’s daughter Michelle in her school play.
Semantic parse: Louis is Robert’s sibling. Michelle is Robert’s child.

Sentence: Bernardo got a cone and Antonio got a sundae.
Semantic parse: None

Sentence: They had a wonderful time.
Semantic parse: None

Sentence: Mary was playing in the sandbox with her brother Dennis.
Semantic parse: Mary is Dennis’s sibling.

Sentence:
Semantic parse: None

Sentence: David is very lucky to have a husband who adores her and treats her like a queen.
Semantic parse: None

Sentence: Dennis id Amy’s only child.
Semantic parse: Dennis is Amy’s child.

Sentence: Michael laughed, and felt better.
Semantic parse: None

Sentence: Angelica ca n’t wait to see her favorite aunt Tracy.
Semantic parse: Tracy is Angelica’s aunt.

Sentence: Marie does n’t like having to babysit her younger brother, Pedro.
Semantic parse: Pedro is Marie’s sibling.

Sentence:
```

Figure 10: Semantic parsing prompt for CLUTRR domain.



Marie and her husband Robert like to go fishing on the weekends.

Marie’s son Allan doesn’t go because he hates fishing.

Which of the following sentences makes the most sense given the information above?

○ Allan and his uncle Robert went to the park.
● Allan and his aunt, Beverly, went to Disney World.

Robert and his brother Antonio played harmonicas together.

Robert’s daughter, Elsie, asked him to play with her.

Which of the following sentences makes the most sense given the information above?

○ Elsie doesn’t like having to babysit her younger brother, Antonio.
● Elsie was playing hide-and-seek with her sister Tracy.

Tracy went to dinner with her daughter Shantel.

Shantel’s daughter, Lizzie, asked her mom to read her a story.

Which of the following sentences makes the most sense given the information above?

○ Lizzie was playing hide-and-seek with her sister Tracy.
● Lizzie loves hanging out with her uncle Antonio.

Figure 11: Additional trials from CLUTRR human judgement experiment (“prompt from model” condition). Selected option was generated by the dual-system model, and the other option was generated by the neural only single-system model.