Exploring Continual Learning of Compositional Generalization in NLI

Xiyan Fu
Dept. of Computational Linguistics
Heidelberg University, Germany
fu@cl.uni-heidelberg.de

Anette Frank
Dept. of Computational Linguistics
Heidelberg University, Germany
frank@cl.uni-heidelberg.de

Abstract

Compositional Natural Language Inference (NLI) has been explored to assess the true abilities of neural models to perform NLI. Yet, current evaluations assume models to have full access to all primitive inferences in advance, in contrast to humans that continuously acquire inference knowledge. In this paper, we introduce the *Continual Compositional Generalization in Inference (C^2Gen NLI)* challenge, where a model continuously acquires knowledge of constituting primitive inference tasks as a basis for compositional inferences. We explore how continual learning affects compositional generalization in NLI, by designing a continual learning setup for compositional NLI inference tasks. Our experiments demonstrate that models fail to compositionally generalize in a continual scenario. To address this problem, we first benchmark various continual learning algorithms and verify their efficacy. We then further analyze C^2Gen, focusing on how to order primitives and compositional inference types, and examining correlations between subtasks. Our analyses show that by learning subtasks continuously while observing their dependencies and increasing degrees of difficulty, continual learning can enhance composition generalization ability.[1]

1 Introduction

Natural Language Inference (NLI) determines the inferential relation between pairs of sentences, by classifying the hypothesis as being true (entailment), undecided (neutral), or false (contradiction) given the premise (Dagan et al., 2013; Bowman et al., 2015; Williams et al., 2018). The task has been researched for decades and has been shown to facilitate downstream NLU tasks such as text summarization (Laban et al., 2022; Utama et al., 2022), question answering (Chen et al., 2021), or dialogue generation (Stasaski and Hearst, 2022).

Recently, large pre-trained models (PLMs) have achieved results on par with human performance by fitting NLI training data (Wang et al., 2019a; Raffel et al., 2020; Chowdhery et al., 2023). Despite the success of state-of-the-art PLMs, it remains unclear to what extent neural models have the ability to generalize when performing NLI. To better assess the true abilities of PLMs to perform NLI, Compositional Generalization (Fodor and Pylyshyn, 1988; Hupkes et al., 2020) evaluation has been proposed for NLI (Yanaka et al. 2020; Geiger et al., 2020; Fu and Frank, 2023). This novel task aims to evaluate whether models are able to predict unseen compositional inferences if they have seen their constituting primitive inferences in training. The left part of Table 1 (Compositional Generalization for NLI) shows an unseen compositional NLI test instance for which we expect a model to make the correct prediction '*He tries to catch his dog ↛ He catches his pet*', by relying on the primitive inferences '*try to S ↛ S*' and '*catch his dog → catch his pet*'.

However, existing work evaluating Compositional Generalization for NLI (*CGen NLI*) relies on offline training, which crucially differs from the way humans acquire knowledge, i.e., by *continual learning* (Ring, 1997; Parisi et al., 2019). Real communication scenarios require the understanding and induction of compositional inferences relative to dynamically updated knowledge. For example, an agent should be able to compose some newly acquired inferential knowledge *buy an apple vision pro (S) → digital content is blended with physical space (S')* with previously learned *try to S ↛ S*, to induce *try to S ↛ S'*. In Section 9, we present a promising application of continual compositional inference in a dialogue setting.

To better align with the compositional generalization ability in real-world situations, and to prepare applying compositional NLI to dynamically

[1]Data and code can be found in https://github.com/Heidelberg-NLP/C2Gen.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 912–932, 2024. https://doi.org/10.1162/tacl_a_00680
Action Editor: Mihai Surdeanu. Submission batch: 2/2024; Revision batch: 4/2024; Published 8/2024.

	Compositional Generalization for NLI (CGen)		Continual Compositional Generalization for NLI (C^{2}Gen)	
train	A man fails to make a snowball ↛ A man plays with a ball A girl tries to do a stunt ↛ A girl performs a bicycle trick	$\mathcal{S}_1$	A girl tries to do a stunt ↛ A girl performs a bicycle trick A man manages to do a stunt ↛ A man performs a bicycle trick	
	A man fails to catch his dog → ¬ A man catches his pet A man manages to do a stunt ↛ A man performs a bicycle trick	$\mathcal{S}_2$	A man fails to catch his dog → ¬ A man catches his pet A man fails to make a snowball ↛ A man plays with a ball	
test	A man tries to catch his dog ↛ A man catches his pet		A man tries to catch his dog ↛ A man catches his pet	

Table 1: NLI train and test instances for Compositional Generalization in a non-continual (CGen) and continual learning (C^{2}Gen) setting. Test instances are unseen compositions, while veridical and NLI inferences have been seen as primitives during training for compositional inference. *C^{2}Gen* simulates human learning via *a continual learning stream*, where one primitive task ($\mathcal{S}_1$) is learned before the other ($\mathcal{S}_2$). In contrast, *CGen* assumes that all data is accessible in advance and *randomly shuffled* for training.

evolving information states, we introduce a new task: *Continual Compositional Generalization for NLI (C^{2}Gen NLI)*, which aims to explore the compositional generalization ability of a model when performing NLI *in a continual learning scenario*. We simulate a continuous learning process by manipulating the order in which specific primitive NLI inferences are encountered during training. The right part of Table 1 shows an example. To solve the unseen compositional inference test sample, a model needs to learn, in the first place, the primitive inference *try to S ↛ S* ($\mathcal{S}_1$), and then *catch his dog → catch his pet* ($\mathcal{S}_2$). The C^{2}Gen NLI task challenges models in two ways: it tests i) their ability to perform *compositional generalization*, by combining learned primitive inferences to solve unseen compositional inferences, and ii) doing this *in a continual learning scenario* that requires models to memorize and re-use primitive inferential knowledge they continually acquired. Unlike the existing CGen NLI task, *C^{2}Gen NLI* allows us to evaluate whether models can learn primitive inferences *continuously* and *efficiently*.

To facilitate research on C^{2}Gen NLI, we establish an evaluation setup and task dataset for systematic analysis of the effect that continual learning has on the compositional generalization capabilities of models to perform NLI. We design two sub-tasks to perform fine-grained compositional generalization analysis: i) *compositional inference (Task$_{CI}$)* to explore how well a model performs compositional inference; and ii) *primitive recognition (Task$_P$)*, to evaluate a model's ability to resolve constituting primitive inferences. With our evaluation datasets and tasks, we conduct experiments in CGen and C^{2}Gen NLI settings using a multi-task model for the different inference tasks. Initial results show that with continual learning, models show inferior performance in compositional NLI inference, which we show to be due to *forgetting*.

To combat the forgetting issue, we benchmark a set of continual learning algorithms targeted at memorization. Our results validate their effectiveness, but also show that memorization alone cannot solve the compositional inference task. To gain deeper understanding of the challenges involved in a continual scenario, we investigate the effect of learning primitive inferences in different orders, analyze correlations between primitive and compositional NLI tasks, and the impact of ordering compositional inference types by difficulty. Our findings highlight the importance of ordering inference types in continual learning according to dependencies and intrinsic difficulty.

Our main contributions are as follows:

i) We motivate and introduce the **C^{2}Gen NLI** (Continual Compositional Generalization for Natural Language Inference) task, which to our knowledge is the first challenge to explore the *compositional generalization* ability of NLI in a *continual learning scenario*.

ii) We construct a *compositional NLI dataset* and rearrange its partitions for C^{2}Gen NLI.

iii) Experiments indicate that *forgetting* is a major challenge for C^{2}Gen NLI. To combat this issue, we benchmark a set of continual learning algorithms and verify their effectiveness.

iv) Further analyses highlight the impact of *guiding the order of continual learning* by

observing *dependencies* and *degrees of difficulty* of primitive and compositional inference types, for compositional NLI performance.

v) By controlling for *data leakage* using pseudo data, we demonstrate that the C^2Gen NLI challenge persists for LLMs such as Llama.

2 Related Work

NLI determines the inferential relation between a hypothesis and a premise (Dagan et al., 2013; Bowman et al., 2015; Lai et al., 2017; Williams et al., 2018; Welleck et al., 2019). Prior work aimed to improve NLI performance with various neural model types (Parikh et al., 2016; Gong et al., 2018; Chen et al., 2018; Bauer et al., 2021). Recently, large PLMs perform well on the NLI task, often achieving human performance (Wang et al., 2019a; Liu et al., 2019). Despite the success of state-of-the-art LLMs, it remains unclear if models are able to generalize when performing NLI. To better assess their inference abilities, research has started to explore to what extent they perform generalization when performing NLI. This includes cross-genre (Williams et al., 2018) and cross-lingual (Conneau et al., 2018) generalization or investigating the impact of heuristics (McCoy et al., 2019; Bhargava et al., 2021). In this work, we evaluate the generalization ability in NLI focusing on compositional generalization. That is, we test a model's capability of predicting unseen compositional inferences if constituting primitive inferences have been learned.

Early work evaluates compositional generalization for NLI targeting novel compositions involving specific linguistic phenomena, e.g., composing predicate replacements and embedding quantifiers (Yanaka et al., 2020), focusing on lexical entailment and negation (Geiger et al., 2020; Goodwin et al., 2020). Recently, Yanaka et al. (2021) and Fu and Frank (2023) extended the scope of compositional generalization evaluation to composition of veridical inference with customary NLI, finding that PLMs are limited in compositionality. Despite promising findings of the above studies, they all assume that models have full access to all training data in advance. This is in contrast with humans acquiring knowledge in a continuous fashion.

To simulate human learning processes, continual learning has been proposed (McCloskey and Cohen, 1989; Wu et al., 2022), enabling models to learn from a continuous data stream over time. Robins (1995) and French (1999) identified catastrophic forgetting being the main challenge in continual learning. To address this issue, various continual learning strategies have been proposed. Among others, data-based algorithms (Chaudhry et al., 2019b,a; Aguilar et al., 2020) are well-known. They use small memories to store seen training data, to be reused in later training steps. Using such strategies, later work designed elaborate models to enhance the performance of tasks such as relation extraction (Wang et al., 2019b), multilingual learning (Berard, 2021; M'hamdi et al., 2023), or dialogue (Madotto et al., 2021). By contrast, we use such continual strategies to analyze the impact of continual learning on compositional generalization ability in NLI.

Both compositional and continual learning are pivotal aspects for evaluating the genuine capabilities of large PLMs. Existing work (Dziri et al., 2023; Berglund et al., 2024; Mitchell et al., 2023) indicates that although LLMs are pre-trained on large amounts of data, they still struggle in novel tasks and situations. Thus, LLMs are expected to learn compositionally and continuously. Some recent work aims to combine continual learning and compositionality. Li et al. (2020) focus on continual learning in a sequence-to-sequence task. They propose to represent syntactic and semantic knowledge separately which allows to leverage compositionality for knowledge transfer. Jin et al. (2020) introduce a challenging benchmark that aims at continual learning of compositional semantics from visually grounded text. Unlike them, we introduce a new task that focuses on Continual Learning of Compositional Generalization in NLI. With this task, we i) analyze the challenge of compositional generalization in NLI in a continual learning setup; ii) identify the effect of ordering primitive and compositional inference types according to their dependencies and difficulty; and iii) finally, in §9 we showcase the relevance of continual learning in NLI in a concrete application, namely, *Persona Dialogue*.

Our finding ii), which highlights the impact of ordering primitive and compositional inference types based on their difficulty, is close to another machine learning paradigm, known as *curriculum learning* (Elman, 1993; Krueger and Dayan, 2009; Bengio et al., 2009; Soviany et al., 2022). This learning paradigm is inspired by the human

classroom, and refers to training a model with a curriculum of increasing difficulty. Existing work first focuses on assessing the difficulty of training samples. According to their difficulty, they further weight data samples and bias the model towards them (Kumar et al., 2010; Huang and Du, 2019), or organize data into subgroups and commence learning from the easiest batch (Xu et al., 2020; Jia et al., 2023; Ranaldi et al., 2023). *Curriculum learning* differs from *continual learning* in two respects:[2] i) *learning schema*. Curriculum learning remains an offline learning method. It focuses on structuring the learning process to facilitate faster and more robust learning, instead, continual learning aims to adapt to new data over time while preserving past knowledge. ii) *training atoms*. Curriculum learning concentrates on data points, instead, continual learning focuses on tasks or knowledge levels. Despite these distinctions, curriculum learning and continual learning interact, such as adopting the ordering principle from curriculum learning to enhance continual learning. Our findings, derived from the analysis of learning sequences in continual learning, could serve as empirical evidence supporting the principles of curriculum learning.

3 Task Setup: C^{2}Generalization in NLI

In this section, we provide an overview of continual learning (§3.1) and describe the construction of our Compositional NLI dataset (§3.2). Building upon this foundation, we rearrange partitions of the dataset to establish *Compositional generalization* tests with standard training (CGen) and a *Continual learning* (C^{2}Gen) setup (§3.3).

3.1 Continual Learning Preliminary

Continual learning (McCloskey and Cohen, 1989; Wu et al., 2022) is proposed to simulate human learning processes, enabling models to learn from a continuous and non in-distribution data stream over time. The objective is to enable a model to continuously learn a set of instances sequentially ordered with respect to a set of n tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$, following a given order. The model is trained on examples from $\mathcal{T}_1$, progresses to $\mathcal{T}_2$, and so on until $\mathcal{T}_n$. Notably, during the learning process for each task $\mathcal{T}_i$, the model is not allowed to access training data from previous tasks $\mathcal{T}_{<i}$ or future tasks $\mathcal{T}_{>i}$. Within each task $\mathcal{T}_i$, instances are trained in a random order. In contrast, conventional training involves full access to all data in advance, meaning the model is trained simultaneously on examples randomly sampled from the set of tasks $\mathcal{T}$.

3.2 Compositional NLI

We model Compositional Inference (CI) building on customary NLI samples. Both customary and compositional NLI involve the relation between premise and hypothesis, but compositional inference involves at least two different primitive inference types (Table 1).[3] To master compositional inference, a model must i) resolve the involved primitive NLI inferences and ii) compose the inferred results, using a suitable composition function.

We construct compositional inferences by selecting *veridical inference* as a special primitive inference type, and combine it with customary NLI inference samples as a second primitive inference (cf. Table 1). Given that veridical inference involves an embedded sentence, it can be flexibly combined and scaled to compositional inference datasets (Yanaka et al., 2021). Veridical inference (Karttunen, 1971; Ross and Pavlick, 2019) is strongly determined by the lexical meaning of sentence embedding verbs. In the context of a factive veridical verb, we can infer that the proposition it embeds can be held to be true, e.g., *He manages to* $S \rightarrow S$. For a non-veridical verb, we cannot infer the truth or falsity of a proposition, e.g., *He tries to* $S \nrightarrow S$; while for non-factive veridical verbs, we can infer the negation of the complement, e.g., *He refuses to* $S \rightarrow \neg S$. For customary NLI we distinguish three classes: e(ntailment): S $\rightarrow$ S', n(eutral): S $\nrightarrow$ S' and c(ontradiction): S $\rightarrow \neg$S'.

To construct *compositional* NLI samples, we ensure that the hypothesis of a (non-)veridical inference pair (*x verb S, S*) matches the premise of a customary NLI pair (S, S'), to derive a transitive inference pair that may be entailed, neutral, or contradictory. For example, *He tries to do* $S \nrightarrow S$ & $S \rightarrow S' \Rightarrow$ *He tries to do* $S \nrightarrow S'$. We use the composition rules listed in Table 2 to define the compositional inference labels. For example, *A man tries to catch his dog* $\nrightarrow$ *A man catches his pet* is a (non-entailing) compositional inference. Here,

[2] We refer to Table 2 in Biesialska et al. (2020) for a more comprehensive comparison.

[3] We restrict ourselves to two primitive components.

index	P^V	P^N	CI
①	positive	entailment	entailment
②	positive	neutral	neutral
③	positive	contradiction	contradiction
④	neutral	entailment	neutral
⑤	neutral	neutral	neutral
⑥	neutral	contradiction	neutral
⑦	negative	entailment	contradiction
⑧	negative	neutral	neutral
⑨	negative	contradiction	entailment

Table 2: Composition rules for compositional inferences. P^V, P^N, and CI indicate veridical, customary and compositional inference, respectively.

tries to S $\nrightarrow$ *S* represents a non-veridical (neutral) inference sample, and *catch his dog* $\rightarrow$ *catch his pet* an entailing inference sample. Composing the above primitive inference results determines the label for the compositional inference, i.e., *neutral* (rule ④).

3.3 Compositional Generalization Testing

Compositional Generalization (CGen) in NLI Compositional generalization tests are designed to evaluate whether a model can generalize to unseen compositional inferences whose constituting primitives have been observed in training. For example, we can evaluate a model's compositional generalization ability by testing it on an unseen compositional sample *A man tries to catch his dog* $\nrightarrow$ *A man catches his pet*, where its constituting primitive inferences *tries to S* $\nrightarrow$ *S* and *catch his dog* $\rightarrow$ *catch his pet* have been seen in training. We denote the set of possible veridical inference types with $\mathcal{V}$, the set of customary inference types with $\mathcal{N}$, and the set of all possible compositional inference types with $\mathcal{C} = \mathcal{V} \times \mathcal{N}$. The domain of all instances of the respective types is given as $D = \{(v, n) | v \in \mathcal{V}, n \in \mathcal{N}, (v, n) \in \mathcal{C}\}$. In all our compositional generalization experiments, we guarantee there is no intersection between the compositional types used in training and test, i.e., $\mathcal{C}_{train} \cap \mathcal{C}_{test} = \emptyset$, while primitive inferences involved in test instances are ensured to have been seen in training: $\mathcal{V}_{test} \subseteq \mathcal{V}_{train}, \mathcal{N}_{test} \subseteq \mathcal{N}_{train}$.

Continual Compositional Generalization (C^2Gen) in NLI Unlike standard compositional generalization evaluation that relies on offline learning, requiring all training data to be processed in advance, the continual compositional generalization test (C^2Gen) extends the evaluation to a continual learning setup. Here, a model is fed with a non-stationary data stream, i.e., the training process follows a controlled learning order, simulating how humans acquire knowledge from their environment. Following the standard CGen setup, we evaluate a model's generalization ability in compositional NLI by testing *unseen* composition types, e.g., *A man tries to catch his dog* $\nrightarrow$ *A man catches his pet*. During training, we separate the training stream into *sequential stages* $\mathcal{S}_i$ ($i \in \{1,2\}$), where i) in one stage the model learns to categorize *veridical inference* based on the embedding verb (e.g., the neutral verb *try*); ii) in the other it learns to categorize a *customary NLI pair* (e.g., the entailment pair *catch his dog* $\rightarrow$ *catch his pet*). Hence, the model first learns one primitive (e.g., $\mathcal{V}$) to solve compositional inference and then the other ($\mathcal{N}$), or vice versa.

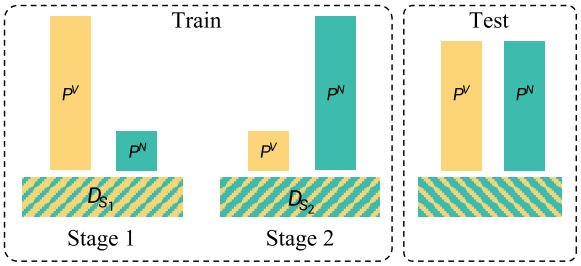

Figure 1: Training and testing setup for compositional inference for continual stages $\mathcal{S}_i$, in C^2Gen. In $\mathcal{S}_1$ we feed various veridicality samples and a few primitive NLI samples. $\mathcal{S}_2$ works analogously.

We construct the above continual scenario by controlling irrelevant variables. When exploring veridical inference in $\mathcal{S}_1$, we use a small number of primitive NLI samples and feed various veridicality samples. Similarly, in $\mathcal{S}_2$, we fix a restricted number of samples from veridical inference and feed various primitive NLI instances. Parallel to training primitives, compositional instances are presented, where the used primitives have been seen in training of the corresponding stage S_i. Different stages are trained sequentially, while samples are randomly trained within each stage. This process enables models to learn incrementally from new data. Figure 1 shows the process.

Compared to customary offline training, C^2Gen NLI is more challenging and innovative. Because models need to learn how to compose primitive inferences, and need to preserve previously acquired knowledge of constituting primitive inferences.

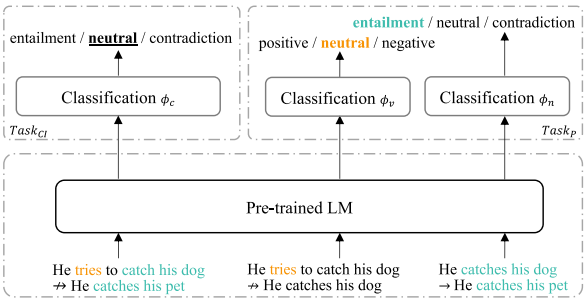

Figure 2: Multi-task architecture for compositional generalization evaluation in CGen & C^{2}Gen NLI. Task$_{CI}$ and Task$_P$ are jointly optimized.

4 Analyzing C^{2}Gen NLI as a Multi-Task

4.1 Decomposing Compositional NLI

To prepare a deep analysis of the generalization capabilities in C^{2}Gen NLI, i.e., compositional NLI in a continual learning training regime, we decompose the CGen task into two constituting subtasks: prediction of primitive inferences (Task$_P$), and prediction of compositional NLI (Task$_{CI}$) as the main task. We apply multi-task learning to jointly learn the two tasks.[4] Figure 2 gives an overview.

Task$_{CI}$: Compositional Inference In the NLI *CI* task, a model is tasked to predict the inferential relationship instantiated in a given compositional NLI sample. For example, the model is expected to predict the value '**neutral**' for *A man tries to catch his dog ↛ A man catches his pet*.[5]

Task$_P$: Primitives Recognition Task$_P$ evaluates whether a model correctly predicts the primitive inferences from which a given compositional sample is built. That is, for *A man tries to catch his dog ↛ A man catches his pet* we test the model predictions for its constituting primitive inferences, expecting i) neutral for *A man tries to S ↛ S* and ii) entailment for the entailed inference *A man catches his dog → A man catches his pet*.

4.2 Model

The **Compositional Inference (Task$_{CI}$)** is defined as a classification task. The model receives as input the concatenation of the premise and the hypothesis of a compositional NLI sample. The model encodes the sequence to a representation x, and we adopt a softmax classifier on top of the classification token of the last layer to predict one of the NLI classes, based on the encoded input representation. We use the cross entropy function to calculate the compositional reasoning loss $\mathcal{L}_{cr}$

$$\mathcal{L}_{cr} = \sum_{(x,c)} \ell_{CE}(\phi_c(x), c) \quad (1)$$

where x is the input representation and c the ground truth label. ϕ_c is the softmax classifier for compositional natural language inference. ℓ_{CE} is the cross-entropy function.

We define the **Primitive Inferences Recognition task (Task$_P$)** as a joint classification task, where each classifier is in charge of a primitive inference. For each primitive inference, we form an input sequence by concatenating premise and hypothesis, and process it in the same way as detailed for the compositional inference task. For the classification, we adopt two softmax classifiers, one for each of the respective tasks. In training we use the cross-entropy function to calculate each primitive's loss. The two primitive losses are jointly considered to train the model for the joint multi-task for primitives recognition $\mathcal{L}_{prim}$

$$\mathcal{L}_{prim} = \sum_{(x,(v,n))} \ell_{CE}(\phi_v(x_v), v) + \ell_{CE}(\phi_n(x_n), n) \quad (2)$$

where x_v, x_n are the respective input representations, (v, n) the corresponding veridical and NLI instances' ground truth labels. ϕ_v, ϕ_n are the softmax classifiers for veridical and natural language inference, and ℓ_{CE} is the cross-entropy function.

In the end, we use the multi-task training strategy to train two tasks, Task$_P$ and Task$_{CI}$. Their objectives $\mathcal{L}_{prim}$ and $\mathcal{L}_{cr}$ are jointly optimized during training, using loss $\mathcal{L} = \mathcal{L}_{prim} + \mathcal{L}_{cr}$.

4.3 Training Settings

Compositional Generalization (CGen). The standard compositional generalization test in NLI relies on *offline training*, where models have full access to all training data in advance. This setup serves as an upper-bound baseline for our experiments. All training data in CGen is

[4] While we expect that task performance will generally profit from MTL with the decomposed subtasks, our main interest is the ability to analyze the effect of continual learning in more detail.

[5] Table 2 shows how the CI NLI value is semantically determined from its constituting NLI primitives.

mixed in a random order. We denote this as $D_{train} = D_{\mathcal{S}_1+\mathcal{S}_2}$.

Continual Compositional Generalization (C^2Gen). This new training setup evaluates the compositional generalization capability in NLI in a continual learning scenario. The model is restricted to follow a *non-stationary data stream*, i.e., all compositional NLI training data is presented in *a specific order* ($D_{train} = D_{\mathcal{S}_1}, D_{\mathcal{S}_2}$).

4.4 Continual Learning Strategies

In order to deeply analyze the challenges of the C^2Gen NLI task, we first benchmark well-known continual learning strategies, designed to combat forgetting. All methods introduce a small, fixed-size so-called *episodic memory*. It consists of samples selected from a previous learning stage, and is used, in a next training stage, in different ways:

Experience Replay (ER). Chaudhry et al. (2019b) utilize samples from a memory directly *for re-training* in future stages. They distinguish three variants: a) ***ER-res(ervoir)*** applies a sampling technique that ensures that each sample has an equal chance of being selected; b) ***ER-buff*** guarantees that the size of the memory at each training stage $\mathcal{S}_i$ is the same; and c) ***ER-mir*** (Aljundi et al., 2019) selects re-training data that is most likely to be forgotten in the next training stage.

Averaged Gradient Episodic Memory (A-GEM). Chaudhry et al. (2019a) constrain the direction of the updated gradient. They calculate the gradient g' of the previous training stage on memory data and project the updated gradient to a direction that is close to g'.

Knowledge Distillation (KD). Aguilar et al. (2020) apply memory samples to distill and preserve knowledge learned in previous stages, by minimizing the difference between the output predictions from the previous stage and the current stage over memory data.

5 Experimental Setup

5.1 Dataset Construction and Verification

We construct datasets with instances chosen from established NLI datasets. i) For *primitive veridical inference*, we select 21 verbs from the dataset of Ross and Pavlick (2019). We restricted our choice to verbs with infinitive complements to ease the construction of compositional samples. Table 3 shows the selected verbs for each class label. ii) For *primitive customary NLI* we extract 2130 instances (e: 747; n: 693; c: 690) from SICK (Marelli et al., 2014), focusing on instances where the inference is based on specific semantic relations including synonymy, hyponymy, active-passive diathesis, etc. For compositional inference, we compose samples from these primitive veridical and customary NLI data points, as described in §3.2. All compositional inferences are categorized into nine groups using the composition rules in Table 2. Table 4 shows the class distribution.[6] The distribution of target class labels (e:n:c) is roughly 1:2:1.

Signature	Instantiations
positive (+)	manage, begin, serve, start, dare, use, get, come
neutral (∘)	hope, wish, expect, try, plan, want, intend, appear
negative (−)	forget, fail, refuse, decline, remain

Table 3: Instantiation of verbs in different signatures used for constructing veridical inference.

type	#num	type	#num	type	#num
①ee_e	5976	④ne_n	5976	⑦ce_c	3735
②en_n	5544	⑤nn_n	5544	⑧cn_n	3465
③ec_c	5520	⑥nc_n	5520	⑨cc_e	3450

Table 4: Distribution of the nine compositional inference types in testing. 'xy_z' specifies the values of the respective primitive inferences types, where 'x' stands for veridical, 'y' for customary NLI, 'z' for compositional inference, with 'x', 'y', 'z' ∈ {entailment(e) / neutral(n) / contradiction(c)}.[4]

As the dataset is automatically constructed from existing datasets, we perform manual human verification to ensure their validity, following Keysers et al. (2020); Liu et al. (2022, 2024). For cost considerations we restricted manual verification to 200 randomly sampled instances. Two annotators specialized in computational linguistics performed this task. They underwent training in practice sessions with direct feedback before starting the annotation process. Their task was to annotate the correct class (entailment, neutral, or contradiction) for each premise-hypothesis pair, for all three inference types. The inter-annotator agreement

[6] Here, we use {e / n / c} to denote the veridical inference types, instead of{positive(p) / neutral(n) / negative(n)}.

calculated by Cohen's kappa was 0.961, 0.954, and 0.917 for the respective inference types.

After the annotation, we computed the consistency between the human-labeled and automatically constructed data for each inference type. Among incorrect *veridical inference* samples (15 cases)[7], 87% of instances are susceptible of a systematic veridicality bias among humans (Ross and Pavlick, 2019). That is, some verbs with neutral signature are often perceived to have positive signature, while our construction follows the semantic definition (cf. Table 2). The remaining 13% are due to a range of different annotation errors. For *customary inference* (based on SICK), we follow the taxonomy of Kalouli et al. (2023) to categorize error samples (24 cases). Applied to our data, the errors are attributed to the following sources: ambiguous (55%), looseness (25%), phrasal verbs (10%), and annotation error (10%). Note that NLI labeling consistency, in general, is still an open issue, relating to factors such as ambiguity and uncertainty (Pavlick and Kwiatkowski, 2019; Nie et al., 2020; Jiang and de Marneffe, 2022). For incorrect *compositional inferences* (25 instances), we note that incorrect primitive inferences cause accumulated errors, accounting for approx. 91.5% of the incorrect compositional inferences. The remaining ones are annotation errors. Still, the consistency for each inference type exceeds 90%, indicating a high quality of our benchmark dataset, which can be a valuable resource for future work.

5.2 Dataset Split

The compositional inference data $\mathcal{D}^C$ is prepared for the **Compositional Generalization in NLI (CGen)** evaluation as follows: Given nine compositional inference types, we conduct nine-fold cross-validation experiments, reporting averaged results. Specifically, each type will once serve as a test dataset $\mathcal{D}^C_{test}$ (e.g., ①), while the remaining eight types are used as training set $\mathcal{D}^C_{train}$ (e.g., ②–⑨). As outlined in §3.3, we guarantee that the primitive inferences used in a given test instance have been seen in training. For Task$_{CI}$ we train on $\mathcal{D}^C_{train}$ and test on $\mathcal{D}^C_{test}$. For Task$_P$ we decompose the instances of $\mathcal{D}^C_{train}$ and $\mathcal{D}^C_{test}$ into their primitive inferences $\mathcal{D}^C_{train} \Rightarrow \mathcal{D}^{P_V}_{train}$ & $\mathcal{D}^{P_N}_{train}$ for primitive recognition training, and test with unseen primitive inferences $\mathcal{D}^{P_V}_{unseen}$, $\mathcal{D}^{P_N}_{unseen}$.

In the **Continual Compositional Generalization in NLI (C^{2}Gen)** setting (cf. §3.3) we maintain the evaluation protocol for both tasks as detailed above for CGen, but split the train set $\mathcal{D}^C_{train}$ into $\mathcal{D}^C_{\mathcal{S}_1}$ and $\mathcal{D}^C_{\mathcal{S}_2}$ s.th. $\mathcal{D}^C_{train} = \{\mathcal{D}^C_{\mathcal{S}_1} \cup \mathcal{D}^C_{\mathcal{S}_2}\}$, and present this data in a continual training stream. For each stage $\mathcal{S}_i$, $i \in \{1,2\}$: i) If it serves to train the model to learn *veridical inference*, we use a small number of NLI samples and feed various veridicality samples. For example, we select data from ②⑤③⑨, where for the pair ②⑤ the model needs to distinguish the effect of positive and neutral veridicality and similarly for ③⑨, where it needs to distinguish the effect of positive and negative veridicality. ii) If the model is tasked to learn *natural language inference*, we use a small number of veridical verbs, selecting data from ④⑥⑦⑧ (for similar reasons as in i). We experiment with alternative data streams, with reversed order in which specific phenomena are trained, once setting $\mathcal{S}_1$ to process training data targeted to $\mathcal{V}$ and $\mathcal{S}_2$ to $\mathcal{N}$, and once choose the opposite assignment to $\mathcal{S}_1$ and $\mathcal{S}_2$. In each stage, we uniformly sample 3200 instances for training.

5.3 Evaluation Metric

We adopt two metrics: i) *Acc(uracy)* reports the percentage of correctly predicted labels for a given task after training on all stages. ii) *Forget* is a commonly used metric in continual learning. It measures to what extent knowledge that a model has learned in $\mathcal{S}_1$ is preserved after training in $\mathcal{S}_2$. For a given task T, *Forget* is calculated as $(Acc_{\mathcal{S}_1}(D^T_{test}) - Acc_{\mathcal{S}_1,\mathcal{S}_2}(D^T_{test}))$ / $Acc_{\mathcal{S}_1}(D^T_{test})$.

5.4 Implementation Details

Backbone. We use RoBERTa-Large[8] (Liu et al., 2019) given its superior NLI performance[9] and training efficiency, following Yanaka et al. (2021) and Fu and Frank (2023). We train using Adam Optimizer with a learning rate 1e-5 and batch size 8.

Continual Learning Strategies. For all evaluations using continual learning strategies, we set the memory size to 100. Following Chaudhry et al. (2019b) and Aljundi et al. (2019), we set the number of replay samples in each step to the batch size

[7] We provide the aggregate count of incorrect samples annotated by two annotators for analysis.

[8] https://huggingface.co/roberta-large.

[9] https://gluebenchmark.com/leaderboard.

for ER-based strategies, including *ER-reservoir*, *ER-buff*, and *ER-mir*. In practice, we add the memory batch to the current batch in training. For fair comparison to other strategies, we set the sample size to be equal to the batch size used for controlling the gradient in AGEM (Chaudhry et al., 2019a) and for distilling knowledge in KD (Aguilar et al., 2020). For each experiment, we perform three runs with different seeds, as in Jin et al. (2020) and Madotto et al. (2021). We report the mean performance with standard deviations in the following experiments.

Hyperparameter Settings. We determine suitable hyperparameters by empirical assessment in a grid search. To assess the impact of the learning rate, we run experiments across a range of learning rates [1e-5, 2e-5, 3e-5] using Adam optimizer.[10] Results indicate that the gap (Δ) between CGen and C^{2}Gen increases monotonically with increasing learning rate, achieving accuracies of [18.11, 19.05, 19.88] for Task$_P$ and [7.44, 8.39, 9.26] for Task$_{CI}$ for the respective choices. We select 1e-5 as the learning rate because its gap is the most negligible compared to the other rates. Moreover, the similarity in gap values between Task$_P$ and Task$_{CI}$ implies that adjusting hyperparameters alone does not significantly impact the subsequent conclusions. We similarly evaluate the impact of memory capacity on continual strategies, for ranges from 2% to 5% of the one-stage training data, corresponding to memory sizes of 50, 100, 150, and 200. Again, the results for the two tasks exhibit a unimodal distribution, with a peak occurring at 100. Therefore, we opt to utilize a memory size of 100.

6 Results and Analysis

6.1 How Does a Model Perform in C^{2}Gen?

We start by analyzing the effects of the different training settings, *CGen* and *C^{2}Gen*, on model performance in the compositional generalization test for NLI (Task$_{CI}$). Table 5 shows the results. In the CGen setting, the model shows decent performance in compositional inference (Task$_{CI}$) with an accuracy of 46.67. Compared to CGen, C^{2}Gen NLI shows a decline for both continual order variants *ver* $\rightarrow$ *nat* and *nat* $\rightarrow$ *ver*, with reductions of 7.27 and 9.31 points, respectively. This suggests that compositional generalization in NLI in a continual learning scenario is more challenging.

[10] We follow Liu et al. (2019) in the selection of potential learning rates, as excessively large or small values can impede convergence in RoBERTa.

Settings	Task$_P$			Task$_{CI}$
	V	N	V+N	
CGen	$99.96_{0.12}$	$94.36_{0.57}$	$94.36_{0.41}$	$46.67_{0.26}$
ver $\rightarrow$ *nat*				
C^{2}Gen ($\mathcal{S}_1$)	$100.00_{0.00}$	–	–	–
C^{2}Gen ($\mathcal{S}_2$)	$80.72_{0.39}$	$94.25_{0.76}$	$76.31_{0.59}$	$39.40_{0.43}$
nat $\rightarrow$ *ver*				
C^{2}Gen ($\mathcal{S}_1$)	–	$93.94_{0.65}$	–	–
C^{2}Gen ($\mathcal{S}_2$)	$99.58_{0.14}$	$71.15_{0.72}$	$70.73_{0.49}$	$37.36_{0.57}$

Table 5: Results for Task$_P$ (incl. individual primitives) and Task$_{CI}$ in different training settings. Subscripts are the standard deviation.

Why is C^{2}Gen More Challenging? To investigate this question, we examine the accuracy of primitive inference (Task$_P$) in different continual learning stages. This is because Task$_{CI}$ is dependent on Task$_P$, requiring correct predictions for the constituting elements of the composition. For C^{2}Gen in order *ver* $\rightarrow$ *nat*, we find that the initially learned veridical primitive inference achieves high accuracy of 100% in stage $\mathcal{S}_1$, showing that the model has achieved perfect knowledge of veridical inference after $\mathcal{S}_1$. However, the accuracy for veridicality drops to 80.72 ($\downarrow$19.18) after learning primitive NLI in $\mathcal{S}_2$. This suggests that the model forgets the primitive knowledge learned during $\mathcal{S}_1$. We find a similar trend in the C^{2}Gen setting *nat* $\rightarrow$ *ver*, where the accuracy of the initially learned NLI primitive inference drops from 93.94 to 71.15 ($\downarrow$22.79). While in each order only one primitive is affected by forgetting, the joint accuracy for Task$_P$ drops to 70-76 points in both settings. From these observations we conclude that **catastrophic forgetting is a major challenge in C^{2}Gen**.

6.2 Can Continual Learning Strategies Help?

Next, we apply existing continual learning strategies that are designed to address the problem of forgetting, and analyze their effect on the preservation of knowledge of primitives (Task$_P$) and on compositional generalization (Task$_{CI}$) in C^{2}Gen, for both learning orders. Table 6 shows the results.

Settings	*ver* → *nat*					*nat* → *ver*				
	Task$_P$				Task$_{CI}$	Task$_P$				Task$_{CI}$
	Acc$_V$(↑)	Acc$_N$(↑)	Acc$_{V+N}$(↑)	Forget$_V$(↓)	ACC(↑)	Acc$_V$(↑)	Acc$_N$(↑)	Acc$_{V+N}$(↑)	Forget$_N$(↓)	ACC(↑)
C^{2}Gen ($\mathcal{S}_2$)	$80.72_{0.39}$	$94.25_{0.76}$	$76.31_{0.59}$	$19.18_{0.39}$	$39.40_{0.43}$	$99.58_{0.14}$	$71.15_{0.72}$	$70.73_{0.49}$	$24.26_{0.48}$	$37.36_{0.57}$
ER - Res	$99.89_{0.01}$	$94.14_{0.56}$	$94.04_{0.53}$	$0.11_{0.01}$	$44.89_{0.68}$	$100.00_{0.00}$	$87.43_{0.67}$	$87.43_{0.67}$	$7.64_{0.53}$	$42.34_{0.71}$
ER - Buff	$99.78_{0.01}$	$94.25_{0.34}$	$93.91_{0.28}$	$0.15_{0.01}$	$44.34_{0.56}$	$100.00_{0.00}$	$87.38_{0.59}$	$87.38_{0.59}$	$6.91_{0.42}$	$41.68_{0.42}$
ER - Mir	$99.92_{0.00}$	$94.87_{0.29}$	$94.04_{0.19}$	$0.08_{0.00}$	$44.73_{0.72}$	$100.00_{0.00}$	$87.55_{0.71}$	$87.55_{0.71}$	$6.71_{0.69}$	$42.01_{0.66}$
AGEM	$99.86_{0.02}$	$94.91_{0.87}$	$93.78_{0.75}$	$0.14_{0.03}$	$42.10_{0.91}$	$99.61_{0.03}$	$81.70_{1.12}$	$81.35_{0.94}$	$13.25_{0.86}$	$41.60_{0.81}$
KD	$99.80_{0.03}$	$94.56_{0.63}$	$90.13_{0.44}$	$0.20_{0.03}$	$42.37_{0.77}$	$97.86_{0.04}$	$82.69_{0.99}$	$81.90_{0.87}$	$11.57_{0.68}$	$41.78_{0.74}$

Table 6: Results of compositional primitive recognition (Task$_P$) and inference (Task$_{CI}$) in C^{2}Gen NLI across different continual learning strategies. Subscripts are the standard deviation.

Compared to vanilla C^{2}Gen, all continual strategies yield improved accuracy for both tasks and reduce the forgetting value of learned primitive inference. C^{2}Gen$_{ver\rightarrow nat}$, yields a significant improvement in the accuracy of the initially learned primitive (Acc$_V$), with an increase from approx. 80 to 100. Accordingly, the forgetting value associated with this primitive decreases by the same amount to almost 0. A similar trend is seen in C^{2}Gen$_{nat\rightarrow ver}$ where the accuracy of the initially learned primitive (Acc$_N$) increases from 71 to 83, while its forget value drops from 24 to 10. This shows that **continual learning strategies alleviate forgetting, helping the model to regain substantive performance** (+5 points for Task$_{CI}$).

We then analyze the effect of different continual strategies. Table 6 shows that Experience Replay strategies (ER-Res/Buff/Mir) achieve superior results with two tasks in different learning orders. For example, in C^{2}Gen$_{ver\rightarrow nat}$ ER-based strategies achieve a Task$_{CI}$ accuracy of 44 (as opposed to 42 for AGEM and KD). With the reverse order C^{2}Gen$_{nat\rightarrow ver}$ the performance is lower for both tasks: Task$_{CI}$ achieves 42 (ER) vs. 41 (non-ER); Task$_P$ yields 87 (ER) vs. 81 (non-ER). The only exception is Task$_{P_V}$ in C^2*Gen*$_{ver\rightarrow nat}$, where all continual strategies show comparable performance, at almost 100%. This is likely due to the ease of learning highly lexicalized veridicality classes, to which continual strategies cannot contribute much (cf. also Table 5).

7 Establishing Learning Order for C^{2}Gen

As shown in §6.2, continual strategies can greatly improve the performance of primitive and compositional NLI in C^{2}Gen NLI. However, the continual learning results still lag behind non-continual training. To gain deeper understanding of the challenges involved in the continual learning process for compositional generalization inference, we perform further analysis of the C^{2}Gen setting.[11]

7.1 Effects of Primitive Learning Orders

While it seems evident that primitive tasks must be learned prior to compositional tasks they are constitutive for, the order among primitive tasks is more difficult to establish. To explore how different orders of learning primitives in continual learning affect compositional generalization, we compare the performance of Tasks P and CI with alternating orders of learning *veridical inference* (*ver*) and *customary NLI inference* (*nat*), i.e., *ver* → *nat* vs. *nat* → *ver*. Table 6 shows that *ver* → *nat* consistently outperforms *nat* → *ver*. For *ER-Res*, e.g., i) for Task$_P$, Acc$_{V+N}$ differs by 6.61 points (94.04 vs. 87.43); ii) for Acc$_{CI}$ in Task$_{CI}$ the difference is smaller, but still considerable (2.55 points). These differences indicate that **the order of learning constituting primitive inferences is relevant for compositional NLI inferences**.

In order to investigate why *ver* → *nat* performs better than *nat* → *ver*, we examine the representation changes of the initially learned primitives for the respective learning orders at different timesteps: i) by the end of $\mathcal{S}_1$, where the model has just completed learning the initial primitive, and ii) after $\mathcal{S}_2$, when the model has completed learning of the other primitive. For $\mathcal{S}_2$ we compare two settings: pure continual learning

[11] In this section we select ER-Res as continual learning strategy for our experiments, given its superior performance (cf. Table 4). The remaining strategies show similar trends.

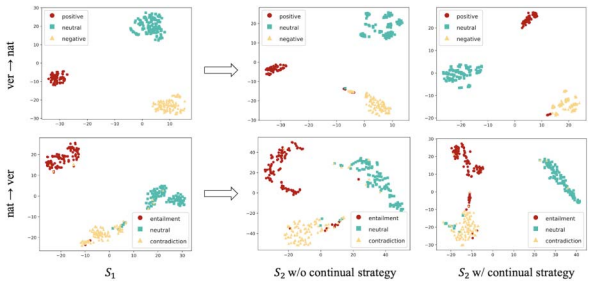

Figure 3: Changes of learned primitive representations from $\mathcal{S}_1$ to $\mathcal{S}_2$ with different learning orders.

($\mathcal{S}_2$ w/o continual strategy) and continual learning using the ER-Res strategy ($\mathcal{S}_2$ w/ continual strategy).

Figure 3 visualizes the results. For both orders, we observe similar changes between $\mathcal{S}_1$ and $\mathcal{S}_2$: The three categories within each primitive inference type are clearly grouped in $\mathcal{S}_1$. In $\mathcal{S}_2$, the shapes of the three clusters get looser in $\mathcal{S}_2$ w/o continual strategy, while with continual strategy in $\mathcal{S}_2$ (rightmost images), the density of each cluster can be recovered. When comparing the density of the individual clusters for the different orders (*ver* $\rightarrow$ *nat* vs. *nat* $\rightarrow$ *ver*), it becomes evident that the clusters in *ver* $\rightarrow$ *nat* exhibit a higher level of density in both stages. This suggests that veridical inference is easier to learn than customary NLI, leading to reduced likelihood of forgetting. This finding highlights the importance of considering the inherent difficulty of learning a primitive, and to **order primitives that are easier to learn first**.

7.2 Continual Learning of Dependent Tasks

To better understand the challenges of compositional NLI in the different learning frameworks, we further analyze the correlation between Task$_P$ and Task$_{CI}$. We decompose the compositional inference testing data into its primitive inferences $\mathcal{D}^{C}_{test} \Rightarrow \mathcal{D}^{P_V}_{test}$ & $\mathcal{D}^{P_N}_{test}$ for primitive recognition. We then categorize all test instances into four groups: i) P(✓)CI(✓), where both tasks yield *correct* predictions. ii) P(✓)CI(✗), where seen primitive inferences are *correctly classified,* but predicting unseen compositions fails. This we identify as *lacking generalization capability*. iii) P(✗)CI(✓) records unseen compositions that are correctly predicted without accurately recognizing their primitives. Given that Task$_P$ is a prerequisite for Task$_{CI}$, this scenario indicates *a shortcut*. iv) For P(✗)CI(✗), where both tasks are incorrectly predicted, the model *fails the complete task*.

Setting Indicates:	P(✓)CI(✓) *correct*	P(✓)CI(✗) *no generalization*	P(✗)CI(✓) *shortcut*	P(✗)CI(✗) *wrong*
CGen	46.05	52.41	0.62	0.92
C^{2}Gen	37.98(Δ8.07)	46.71(Δ5.70)	1.42(Δ0.80)	13.89(Δ12.97)
ER-Res	44.33(Δ1.72)	54.38(Δ1.97)	0.56(Δ0.06)	0.73(Δ0.19)

Table 7: Distribution of class performance across Task$_{P\times CI}$ for different settings (all: *ver* $\rightarrow$ *nat*). Δ indicates the gap compared to CGen.

Table 7 displays the distribution of these cases. For CGen, we find an exceedingly low percentage of instances in the P(✗)CI(✓) category, indicating a scarcity of model shortcuts. Since P(✓)CI(✓) and P(✓)CI(✗) jointly cover the remaining probability mass, we conclude that the model meets the preconditions for solving Task$_{CI}$ by being able to solve Task$_P$. But, about half of these cases fail to perform compositional NLI inference in Task$_{CI}$. This suggests that evaluated models lack compositionality. In contrast, human annotation evaluations (§5.1) show that incorrect compositional inferences mainly stem from accumulated errors in primitive inferences. That is, P(✗)CI(✗) is more predominant compared to P(✓)CI(✗). This indicates that humans show greater proficiency in handling compositionality compared to models.

Continual learning in C^{2}Gen shows a reduction in the proportions of P(✓)CI(✓) and P(✓)CI(✗), with the majority of erroneous predictions transitioning to P(✗)CI(✗). This shows that continual learning has a clear impact on primitive recognition, with or without generalization ability. Enhancing the model with strategy ER-Res yields a reduction for P(✗)CI(✗) and a corresponding increase of the P(✓)CI(✓) and P(✓)CI(✗) classes. However, the increase is more pronounced for the *no generalization class* (+3.7). That is, ER-Res proves more effective for primitives compared to compositional generalization. This may be due to the complexity of two tasks, making it relatively easier for primitives to recover from forgetting. Overall, we show that memorization methods can alleviate the forgetting effect for primitives, while compositional inference remains challenging, with a small decrease compared to CGen.

7.3 C^{2}Gen by Increasing Difficulty of Tasks

As the above analysis shows, C^{2}Gen remains challenging with a gap of Δ1.78 for $CI^{ER-Res}_{ver\rightarrow nat}$ compared to CGen (Table 6). We aim to explore how to relieve this issue. Inspired from our insights

	N_e	N_n	N_c	Avg.CI_V	Function Types
V_e	19.50	73.86	13.91	35.76	$f_{v_e}(v_e,X) = X$
V_n	100	100	57.21	85.74	$f_{v_n}(v_n,\_) = n$
V_c	13.99	26.50	15.03	18.51	$f_{v_c}(v_c,X) = \neg X$
Avg. CI_N	44.50	66.79	28.72	46.67	

Table 8: Task$_{CI}$ accuracy (CGen) for all inference types. V, N denote veridical and customary inference. Avg. states average results for different function types. Color indicates the CI target labels: (entailment), (neutral), (contradiction).

into ordering effects for primitive inference types (§7.1) and the curriculum learning paradigm, we investigate the effect of ordering the continual learning stream for the complete compositional task along the degree of difficulty for all involved NLI types.

Table 8 shows that the 9 compositional inference types can be grouped into 3 function types based on veridicality:[12] i) for positive verbs v_e, the compositional inference label is consistent with the label of the NLI primitive; ii) for neutral verbs v_n, compositional inference remains neutral regardless of the NLI inference type; iii) for negative verbs v_c, the compositional inference label is the inverse of the customary NLI label. The respective function types f_{v_x} are defined in Table 8. We determine the difficulty of the individual functions by averaging the results of the individual inferences pertaining to each veridicality label x in the CGen setup. Table 8 shows that the performance of the 3 functions varies considerably: f_{v_n}, for neutral veridicality, exhibits significantly higher accuracy (85.74) compared to the other ones; f_{v_e} for positive veridicality performs much worse (35.76) but still better than f_{v_c} for negative veridicality, with 18.51 points. We hence define two *compositional function learning orders (cfo)* for Task$_{CI}$: *easy* $\rightarrow$ *hard*: $f_{v_n}f_{v_e}f_{v_c}$ and *hard* $\rightarrow$ *easy*: $f_{v_c}f_{v_e}f_{v_n}$.

Following $\mathcal{S}_2$ of the learning process as of §7.2, we add a stage $\mathcal{S}_3$ that only presents compositional inference training data, controlled by a continual data stream where the functions f_{v_e}, f_{v_n}, f_{v_c} are arranged by degree of difficulty. $\mathcal{S}_{3,cfo}$ in Table 9 shows the results of C^{2}Gen in the two opposing orders. For fair comparison, CGen is also trained with this data, yet in random order, achieving 48.64 accuracy. Indeed, applying the *easy* $\rightarrow$ *hard* learning order narrows the gap to CGen up to a small margin of Δ0.42, outperforming *hard* $\rightarrow$ *easy* considerably (Δ2.58). This finding indicates that **further training with a favorable function learning order benefits C^{2}Gen**, aligning with our insight from §7.2, that learning easy components first enhances learning performance.

[12] We take veridicality as example; NLI works analogously.

	CGen	C^{2}Gen easy$\rightarrow$ hard: $f_{v_n}f_{v_e}f_{v_c}$	C^{2}Gen hard$\rightarrow$ easy: $f_{v_c}f_{v_e}f_{v_n}$
$\mathcal{S}_{2,v\rightarrow n}$	46.67	44.89 (Δ 1.78)	
$\mathcal{S}_{3,\ cfo}$	48.64	48.22 (Δ0.42)	46.06 (Δ2.58)
$\mathcal{S}_3^{p\ \mid\ c_{cfo}}$	47.19	45.45 (Δ1.74)	44.63(Δ2.56)

Table 9: C^{2}Gen Task$_{CI}$ accuracy with *compositional function ordering* in two $\mathcal{S}_3$ settings (row 2–3). $\bar{\Delta}$ shows the gap to $CGen$ for respective stages $\mathcal{S}_i$.

	words replacement (natural word: artifical word)
ver	manage: blicke, begin: dmaop, hope: lugi, wish: fepo, expect: kikioa, fail: mfkd, refuse: qneopl
nat	coat-jacket: nlnx-walhra, person-man: fibqpc-qpj, rapidly-quickly: sxaokpw-zssgjuk, dog-pet: ozf-yqj, small-big: noquz-srv, wet-dry: xiw-vcs,

Table 10: Examples of pseudo words for *ver*idical verbs and semantically related terms for *nat*.

To further consolidate the above finding we conduct a complementary experiment $\mathcal{S}_3^{p\ \mid\ c_{cfo}}$. Here, we construct a learning scheme that follows *easy before hard* but strictly orders *primitive before compositional* inference. That is, the model is forced to learn independent primitive inference first, and later compositional inferences ordered by function difficulty. Row 4 in Table 9 indicates that *easy* $\rightarrow$ *hard* still improves over the reverse order, confirming the *easy before hard* scheme. We also note that $\mathcal{S}_3^{p\ \mid\ c_{cfo}}$ yields a larger gap compared to $\mathcal{S}_{3,cfo}$ (1.74 vs. 0.42). This suggests learning CI in parallel to P in $\mathcal{S}_1$, $\mathcal{S}_2$ is beneficial.

8 Controlling Model Size & Data Leakage

PLMs (Devlin et al., 2019; Liu et al., 2019) have demonstrated impressive performance on many NLP tasks through pre-training on extensive data. Recent advancements in large PLMs (Chowdhery et al., 2023; Touvron et al., 2023) have achieved even more substantial improvements by further scaling models and data. However, this raises concerns regarding the reliability of generalization

	Settings	original				pseudo			
		Task$_P$			Task$_{CI}$	Task$_P$			Task$_{CI}$
		V	N	V+N		V	N	V+N	
	CGen	100.00	92.92	92.92	46.15	91.76	83.55	81.37	39.34
ver → *nat*	C^{2}Gen ($\mathcal{S}_1$)	100.00	–	–	–	90.14	–	–	–
	C^{2}Gen ($\mathcal{S}_2$)	81.29(Δ18.71)	92.48	78.83	37.98(Δ8.17)	76.29 (Δ13.85)	81.57	73.82	36.62(Δ2.72)
nat → *ver*	C^{2}Gen ($\mathcal{S}_1$)	–	93.15	–	–	–	82.19	–	–
	C^{2}Gen ($\mathcal{S}_2$)	99.87	73.91(Δ19.24)	72.42	34.64 (Δ11.51)	89.72	59.42 (Δ22.77)	56.72	33.91 (Δ5.43)

Table 11: Performance of Task$_P$ and Task$_{CI}$ on original vs. pseudo dataset in different training settings.

evaluations: i) *re. data*: whether evaluation data might have been encountered during pre-training; ii) *re. model scale*: whether a scaled PLM could show emerging compositional generalization ability. We address these concerns in two experiments.

8.1 Controlling for Data Leakage

Following Lake and Baroni (2023), we construct a *pseudo-compositional* inference dataset by replacing all relevant knowledge-bearing natural language terms with pseudo-words. For veridical inference we replace veridical verbs with pseudo words, e.g., manage → blicke. Table 10 shows examples. Irrespective of these applied changes, we leave the signatures of the original verbs untouched. For customary NLI, we replace pairs of semantically related words that are crucial for deciding the NLI class with a pair of pseudo words. For example, in '*A man catches his dog → A man catches his pet*' we replace *dog → ozf* and *pet → yqj*. Given the difficulty of identifying crucial semantic relations in the NLI data, we select 438 relation pairs covering 813 NLI instances (again examples in Table 10). Like veridical inference, we preserve the original inference labels. Using these *pseudo* primitive inference indicators, we build a *pseudo* Task$_{CI}$ dataset following the process in §3.2.

With this pseudo dataset we re-evaluate the performance of RoBERTa under CGen and C^{2}Gen. The results in Table 11 align well with the trends we have seen in Table 5, for the same data in natural language. This shows that the results of our generalization experiments are not affected by data seen in pre-training. Indeed, compared to CGen, C^{2}Gen NLI shows a decline for both continual order variants of primitives *ver* and *nat*, in both datasets. This confirms that compositional generalization in NLI is more challenging in a continual learning setup. Comparing Task$_P$ and Task$_{CI}$ with alternating orders, we note that *ver → nat* outperforms *nat → ver* in both datasets.

	Settings	Task$_P$			Task$_{CI}$
		V	N	V+N	
	CGen	100.00	95.17	95.17	49.51
nat → *ver*	C^{2}Gen ($\mathcal{S}_1$)	100.00	–	–	–
	C^{2}Gen ($\mathcal{S}_2$)	82.89(Δ17.11)	95.08	79.63	44.39(Δ5.12)
ver → *nat*	C^{2}Gen ($\mathcal{S}_1$)	–	95.14	–	–
	C^{2}Gen ($\mathcal{S}_2$)	99.43	75.24(Δ19.90)	74.82	42.63(Δ6.88)

Table 12: Results for Task$_P$ and Task$_{CI}$ in different training settings with Llama2-7b.

Finally, we observe that the absolute accuracies obtained for Task$_P$ and Task$_{CI}$ on *pseudo* data generally drop compared to the original data, and substantially so for Task$_P$. As for the relative performance of different continual orders regarding *ver* and *nat* in Task$_P$, we note that the relative drop for *nat → ver* compared to its opposite is much more pronounced for *pseudo* vs. original data.

8.2 Model Scale: Testing C^{2}Gen with Llama

We next test the generalization ability for C^{2}Gen NLI for a large PLM such as Llama-2-7B (Touvron et al., 2023). Table 12 shows the results. To fine-tune this large PLM, we adopt standard parameter-efficient fine-tuning (peft) with LoRA (Hu et al., 2022). Compared to RoBERTa-Large, its size increases by approx. 20 times from 0.355 to 7 billion parameters. This enhances the accuracy on the CGen test from 46.67 to 49.51% (Δ 2.84) for Task$_{CI}$. While this marks a progress, a large drop occurs for continual learning in C^{2}Gen (Δ 5.12/Δ 6.88). This suggests that compositional generalization is still a challenge for LLMs. Besides, the gain of 2.84 over RoBERTa on CGen

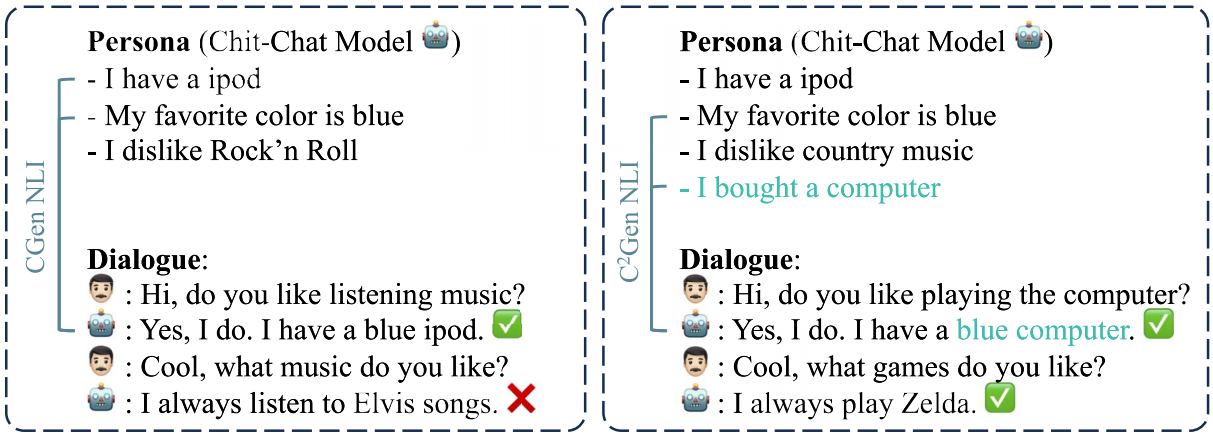

Figure 4: *Persona Dialogue* application for C^{2}Gen: NLI verifies the consistency of dialogue turns generated from *dynamically updated* persona information. We show a profile with *new information* and compositional inferences using it (in ■).

is constrained, compared to the substantial resource cost. This finding is consistent with Qiu et al. (2022), who found that fine-tuning LLMs generally has a flat or negative scaling curve on compositional generalization in semantic parsing.

Similar to RoBERTa, we observe that Llama-2 is affected by forgetting – but the amount of forgetting in Llama-2 does not differ much, dropping by 1.6 points in *ver* → *nat* but rising by 0.66 points in *nat* → *ver*. Comparing different training orders (*ver* → *nat*, *nat* → *ver*) confirms that Llama-2 also benefits from an 'easy to hard' learning scheme.

9 Potential Applications

Our work introduces the new C^{2}Gen NLI task as a first step to explore the compositional generalization ability of models performing NLI in a continual learning setup. Similar to existing continual NLP-based tasks (Wang et al., 2019b; Berard, 2021; Madotto et al., 2021; M'hamdi et al., 2023), the continual learning setup inspires models to learn new inference knowledge continuously, to avoid costs for model retraining. Given such capabilities, the C^{2}Gen NLI task setting can benefit future applications that require the understanding and induction of compositional inferences relative to dynamically updated knowledge stores.

We use the widely researched task *Personalized Dialogue Agent (PDA)* (Zhang et al., 2018) as an example to show how the C^{2}Gen NLI task could apply in a dynamic setting. *PDA* proposes chit-chat models that are conditioned on information provided in a given personality profile. Figure 4 shows an illustration. Existing approaches suffer from consistency issues when a chit-chat model generates utterances that contradict their personality profile. For example, *I dislike Rock'n Roll* contradicts *I always listen to Elvis songs*. To solve this issue, some studies (Welleck et al., 2019; Utama et al., 2022) proposed to use NLI to evaluate and improve consistency. We can achieve this by evaluating whether the persona information entails or contradicts a dialogue utterance. In dialogue, utterances show semantic composition effects when combining primitive information to form new and meaningful sentences. For example, *I have a blue iPod* composes information from *I have an iPod* and *my favorite color is blue*. This scenario aligns with the CGen NLI setup.

But the persona profile of a chit-chat is dynamic and gets updated over time. For example, Fig. 4 shows persona information that is updated with a fact on a new product *computer*. The new primitive can be composed with previously learned primitives to generate novel compositional facts, e.g., *I have a blue computer* from *I bought a computer* and *my favorite color is blue*. Here, re-training the model to update the profile's information state is expensive and time-consuming. By contrast, enabling the model to perform continual learning is a more viable and economic solution. The model is then deemed to evaluate compositional inferences *relative to the updated information state*, aligning with our new task C^{2}Gen NLI.

10 Conclusions and Future Work

We propose C^{2}Gen, a new challenge task for compositional generalization in NLI, grounded in a continual learning scenario. Our new task

targets NLP applications that rely on composing information from continuously updated sources.

By conducting rich analyses for this novel task, on our new benchmark, we show that in continual learning, neural models fail to generalize to unseen compositional inferences due to *forgetting*. With known continual learning strategies we can combat forgetting, but our analyses show that *memorization alone cannot solve the compositional inference challenge*. Our in-depth analyses of C^2Gen show that the model benefits from *learning primitive before compositional inference*, and *learning easy before hard inference subtasks*.

Our findings highlight the importance of observing differences of primitive and compositional inference types, and establishing the relative difficulties of diverse primitive and compositional inference types. With this, we establish recipes that can improve continual learning to approach non-continual learning. New methods can determine optimal learning orders for diverse inference types, while ensuring sufficient diversity in the data stream. Our insights could also benefit other compositional generalization methods, e.g., by ordering demonstrations in in-context learning along principles we established to improve compositional generalization in continual learning.

Acknowledgments

We are grateful to the anonymous reviewers, and action editors Mihai Surdeanu and Katrin Elisabeth Erk for their valuable comments. This work has been supported through a scholarship provided by the Heidelberg Institute for Theoretical Studies gGmbH.

References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7350–7357. https://doi.org/10.1609/aaai.v34i05.6229

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. 2019. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems 32*. https://doi.org/10.1109/CVPR.2019.01151

Lisa Bauer, Lingjia Deng, and Mohit Bansal. 2021. ERNIE-NLI: Analyzing the impact of domain-specific external knowledge on enhanced representations for NLI. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 58–69, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.deelio-1.7

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. https://doi.org/10.1145/1553374.1553380

Alexandre Berard. 2021. Continual learning in multilingual NMT via language-specific embeddings. In *Proceedings of the Sixth Conference on Machine Translation*, pages 542–565, Online. Association for Computational Linguistics.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on ‘‘a is b’’ fail to learn ‘‘b is a’’. In *International Conference on Learning Representations*.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.insights-1.18

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.574

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1075

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip H. S. Torr, and Marc' Aurelio Ranzato. 2019b. On tiny episodic memories in continual learning. *arXiv: Learning*.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.324

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1224

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1269

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220. https://doi.org/10.1007/978-3-031-02151-0

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith

and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99. https://doi.org/10.1016/0010-0277(93)90058-4, PubMed: 8403835

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71. https://doi.org/10.1016/0010-0277(88)90031-5, PubMed: 2450716

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135. https://doi.org/10.1016/S1364-6613(99)01294-2, PubMed: 10322466

Xiyan Fu and Anette Frank. 2023. SETI: Systematicity evaluation of textual inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4101–4114, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.252

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.blackboxnlp-1.16

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *International Conference on Learning Representations*.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.177

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 389–398, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1037

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795. https://doi.org/10.1613/jair.1.11674

Qi Jia, Yizhu Liu, Haifeng Tang, and Kenny Zhu. 2023. In-sample curriculum learning by sequence completion for natural language generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11937–11950, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.666

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374. https://doi.org/10.1162/tacl_a_00523

Xisen Jin, Junyi Du, Arka Sadhu, Ram Nevatia, and Xiang Ren. 2020. Visually grounded continual learning of compositional phrases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2018–2029, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.158

Aikaterini-Lida Kalouli, Hai Hu, Alexander F. Webb, Lawrence S. Moss, and Valeria De Paiva. 2023. Curing the sick and other

NLI maladies. *Computational Linguistics*, 49(1):199–243. https://doi.org/10.1162/coli_a_00465

Lauri Karttunen. 1971. Implicative verbs. *Language*, 340–358. https://doi.org/10.2307/412084

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Kai A. Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394. https://doi.org/10.1016/j.cognition.2008.11.014, PubMed: 19121518

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems*, 23.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177. https://doi.org/10.1162/tacl_a_00453

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural language inference from multiple premises. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 100–109, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Brenden M. Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, 1–7.

Yuanpeng Li, Liang Zhao, Kenneth Church, and Mohamed Elhoseiny. 2020. Compositional language continual learning. In *International Conference on Learning Representations*.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022. Challenges in generalization in open domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-naacl.155

Wei Liu, Stephen Wan, and Michael Strube. 2024. What causes the failure of explicit to implicit discourse relation recognition? *arXiv preprint arXiv:2404.00999*.

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*.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467. Online and Punta Cana, Dominican Republic, Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.590

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier. https://doi.org/10.1016/S0079-7421(08)60536-8

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,

pages 3428–3448, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1334

Meryem M'hamdi, Xiang Ren, and Jonathan May. 2023. Cross-lingual continual learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3908–3943, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.217

Melanie Mitchell, Alessandro B. Palmarini, and Arsenii Kirillovich Moskvichev. 2023. Comparing humans, GPT-4, and GPT-4v on abstraction and reasoning tasks. In *AAAI 2024 Workshop on ''Are Large Language Models Simply Causal Parrots?''*.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.734

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1244

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71. https://doi.org/10.1016/j.neunet.2019.01.012, PubMed: 30780045

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694. https://doi.org/10.1162/tacl_a_00293

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022. Evaluating the impact of model scale for compositional generalization in semantic parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9157–9179, Abu Dhabi, United Arab Emirates, Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.624

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Leonardo Ranaldi, Giulia Pucci, and Fabio Massimo Zanzotto. 2023. Modeling easiness for training transformers with curriculum learning. Ruslan Mitkov and Galia Angelova, editors, In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 937–948, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria. https://doi.org/10.26615/978-954-452-092-2_101

Mark B. Ring. 1997. Child: A first step towards continual learning. *Machine Learning*, 28:77–104. https://doi.org/10.1023/A:1007331723572

Anthony Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146. https://doi.org/10.1080/09540099550039318

Alexis Ross and Ellie Pavlick. 2019. How well do NLI models capture verb veridicality? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2230–2240, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1228

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565. https://doi.org/10.1007/s11263-022-01611-x

Katherine Stasaski and Marti Hearst. 2022. Semantic diversity in dialogue with natural language inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 85–98, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.6

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Prasetya Utama, Joshua Bambrick, Nafise Moosavi, and Iryna Gurevych. 2022. Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.199

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019b. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1086

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1363

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1101

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.542

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.543

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. Exploring transitivity in neural NLI models through veridicality. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 920–934, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.78

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1205