Dataset Cartography for Compositional Generalization

by

Osman Batur İnce

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

KOÇ ÜNİVERSİTESİ

July 22, 2024

Dataset Cartography for Compositional Generalization

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Osman Batur İnce

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Aykut Erdem (Advisor)

_____

Prof. Tunga Güngör

_____

Assist. Prof. Gözde Gül Şahin

Date: _____

To my family

ABSTRACT

Dataset Cartography for Compositional Generalization
Osman Batur İnce
Master of Science in Computer Science and Engineering
July 22, 2024

Neural networks have revolutionized language modelling and excelled in various downstream tasks. However, the extent to which these models achieve compositional generalization comparable to human cognitive abilities remains debatable. While existing approaches in the field have mainly focused on novel architectures and alternative learning paradigms, we introduce a pioneering method harnessing the power of dataset cartography [Swayamdipta et al., 2020]. By strategically identifying a subset of compositional generalization data using this approach, we achieve a remarkable improvement in model accuracy, yielding enhancements of up to 10% on CFQ and COGS datasets. Notably, our technique incorporates dataset cartography as a curriculum learning criterion, eliminating the need for hyperparameter tuning while consistently achieving superior performance. Moreover, as the data becomes the bottleneck in the current large language model (LLM) pipeline, covering every possible combination of known words or phrases becomes infeasible. Therefore, we focus on compositional generalization in LLMs to help LLMs process the combinations of unseen language parts faithfully. We expand the previously described setting above to LLMs and propose a new diversity-aware subset selection method named DiCart, a fusion of dataset cartography and determinantal point processes. DiCart results in better or on-par compositional generalization than baselines and even the full training set.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Osman Batur İnce
Bilgisayar Mühendisliği, Yüksek Lisans
22 Temmuz 2024

Sinir ağları dil modellemeyi devrim niteliğinde değiştirerek çeşitli ardıl görevlerde üstün başarı göstermiştir. Ancak, bu modellerin insan bilişsel yeteneklerine benzer bileşimsel genelleme elde etme derecesi tartışmalıdır. Alandaki mevcut yaklaşımlar ağırlıklı olarak yeni mimarilere ve alternatif öğrenme paradigmalarına odaklanmışken, biz veri kümesi haritalamanın gücünden yararlanan öncü bir yöntem tanıtıyoruz [Swayamdipta et al., 2020]. Bu yaklaşımı kullanıp bileşimsel genelleme verilerinin bir alt kümesini stratejik olarak belirleyerek, model doğruluğunda dikkate değer bir iyileşme sağladık ve CFQ ve COGS veri kümelerinde %10'a varan gelişmeler elde ettik. Özellikle, tekniğimiz veri kümesi haritalamayı bir müfredat öğrenme kriteri olarak da içererek hiperparametre ayarlamasına gerek kalmadan sürekli olarak üstün performans elde edilmesini sağlıyor. Ayrıca, veri mevcut büyük dil modeli (BDM) çerçevesinde darboğaz haline geldiğinden, bilinen kelime veya ifadelerin her olası kombinasyonunu kapsamak imkansız hale gelmektedir. Bu nedenle, BDM'lerin bilinmeyen dil parçalarının kombinasyonlarını doğru bir şekilde işlemelerine yardımcı olmak için bileşimsel genellemeye odaklanıyoruz. Az önce tanımlanan çalışmayı BDM'lere genişletiyor ve veri kümesi haritalama ile determinant nokta süreçlerinin bir birleşimi olan yeni bir çeşitlilik farkındalığına sahip alt küme seçme yöntemi olan DiCart'ı sunuyoruz. DiCart, kıyaslanan tekniklere ve hatta tam eğitim kümesine kıyasla daha iyi veya benzer bileşimsel genelleme performansı sağlıyor.

ACKNOWLEDGMENTS

I want to start by thanking my advisor Assoc. Prof. Aykut Erdem due to his unshakeable support, mastery of research skills, and for opening innumerable gates for me to improve myself beyond I ever imagined. I would like to expand these acknowledgements to Prof. Erkut Erdem and thank him for inspiring me to keep pushing on, no matter how tight everything seems. I want to thank Semih Yagcioglu for his immense help in jump-starting my multimodal learning career. I acknowledge KUIS AI Center for its support via the fellowship.

I would like to thank "Nihai Konsey" for transforming life in İstanbul into one of the most joyous experiences of my life with football matches, watching the World Cup and Euros, playing basketball, lab banter, and just being together. I express my gratitude to Recep for being my first friend and roommate in Koç, and for sharing countless experiences.

Above all else, I wholeheartedly appreciate the support of my family. I appreciate my twin for being more than my best friend, my elder brother for his broad vision, my dad for helping me understand how academia works, and my mom for listening to my rants.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

NLP	Natural Language Processing
OOD	Out-of-distribution
NLI	Natural Language Inference
seq2seq	Sequence-to-sequence
Inv PPL	Inverse Perplexity
OOV	Out-of-vocabulary
CL	Curriculum Learning
LLM	Large Language Model
DPP	Determinantal Point Process
DiCart	Diverse CARTography
SMCalFlow-CS	SMCS

Chapter 1

INTRODUCTION

In recent years, deep learning methods and machine learning infrastructure have made remarkable progress, enabling models to surpass human-level performance in numerous tasks. Natural language processing (NLP) is at the forefront of this progress. Models based on Transformers [Vaswani et al., 2017] such as BERT [Devlin et al., 2019] and benchmarks like SuperGLUE [Wang et al., 2019] led to significant advancements in language modelling and various downstream tasks. However, there is an ongoing debate on whether these models exhibit compositional generalization [Fodor and Pylyshyn, 1988, Smolensky, 1988, Marcus, 2001, Lake and Baroni, 2017].

Compositional generalization refers to the ability of a model to combine known parts of a sentence, such as primitive tokens, to generate novel compositions of these primitive elements. It is considered a fundamental aspect of human cognition and linguistics [Fodor and Pylyshyn, 1988]. Compositional generalization is crucial for enhancing the robustness and practical use of deep learning models in addition to their human aspect. Efforts to understand and improve the compositional generalization abilities of models have gained significant attention lately. Researchers have recently explored techniques such as compositional data augmentation [Andreas, 2020, Qiu et al., 2022], meta-learning [Lake, 2019], and structural priors [Russin et al., 2020]. Additionally, the importance of architectural modifications to capture compositional structures more effectively, such as attention mechanisms [Li et al., 2019] and hierarchical structures [Weißenhorn et al., 2022] have been investigated recently. In another direction, there is also an increasing interest in studying the compositional generalization abilities of Transformers [Ontanon et al., 2022, Csordás et al., 2021, Dziri et al., 2023].

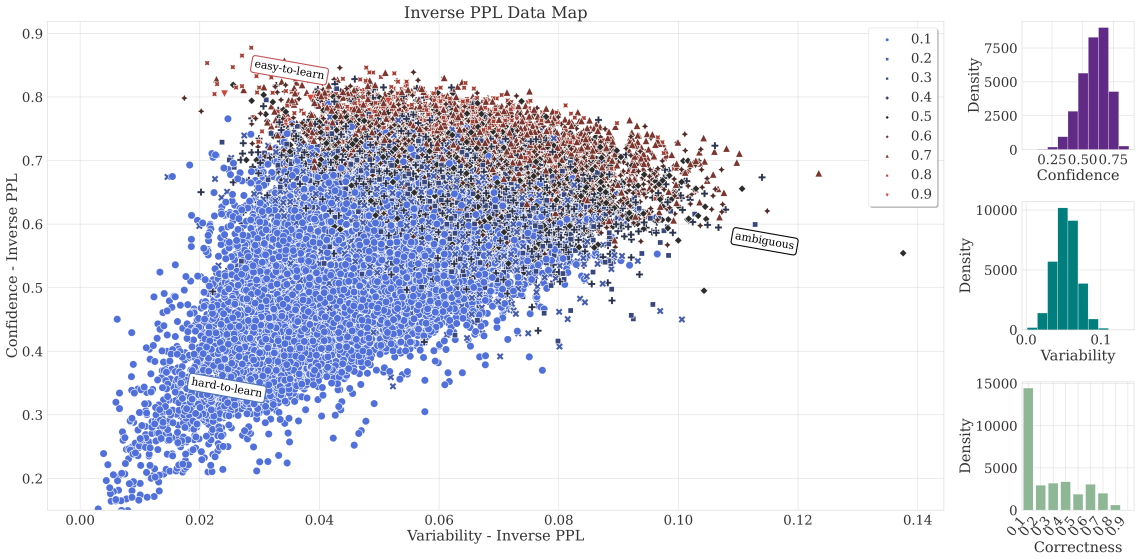

Figure 1.1: **Data map of CFQ train set** for the Transformer model based on Inv PPL measure (converge epoch 20). We expand data maps to generative tasks with different measures and show their contribution to compositional generalization.

In this thesis, we take a distinct approach and harness the power of *dataset cartography* [Swayamdipta et al., 2020] to explore how training dynamics can improve the compositional generalization abilities of neural models ranging from Bi-LSTM to vanilla Transformer to pre-trained Transformer language models. Dataset cartography is a recently proposed technique that quantifies the variability and confidence associated with instances during training, capturing their ambiguity and difficulty, thereby representing the *informational value* of each training sample. [Swayamdipta et al., 2020] demonstrated that it could be used to improve out-of-distribution (OOD) generalization in models for classification-based natural language inference (NLI) tasks. As compositional generalization is inherently an OOD task, we hypothesize that harnessing dataset cartography in compositional generalization can provide new insights.

We establish two experimental setups to investigate the dataset cartography for compositional generalization. These setups are chronological, meaning we started looking into the second setup after successfully examining the first setup.

In the first setup, we diverge from the original cartography setup and focus on

language generation tasks for the systematic generalization problem. We propose an experimental setting to apply dataset cartography to a generative task (see Figure 1.1). Initially, we train a sequence-to-sequence (seq2seq) Transformer model from scratch using the complete training set for only a few epochs. Throughout the training, the dynamics of each instance are observed and recorded separately. Next, we utilize these stored training dynamics to create a reduced training set by selecting specific samples to train the model or build a curriculum by gradually unlocking unobserved examples throughout the training.

Our experimental setup has notable challenges beyond the compositional generalization setting, distinguishing it from the setup originally used in [Swayamdipta et al., 2020]. Instead of relying on crowdsourced datasets that are prone to errors and heavily reliant on data quality, we utilize synthetically generated datasets, namely CFQ [Keysers et al., 2020] and COGS [Kim and Linzen, 2020], which are free from such limitations. Moreover, these datasets are relatively smaller in size, making it challenging to achieve performances *on par* with the 100% train set when using smaller subsets. Lastly, unlike [Swayamdipta et al., 2020], we tackle the complexity of learning the task directly without using pre-trained models. This becomes even more pronounced as the datasets contain non-natural language, rendering pre-training less applicable and learning much harder. Finally, and more importantly, as we are dealing with language generation tasks, quantifying how hard a sequence is, is not straightforward. To address this, we base our evaluation by utilizing inverse perplexity (Inv PPL), CHIA [Bhatnagar et al., 2022], and BLEU [Papineni et al., 2002] as confidence measures, avoiding the overly strict exact matching strategy.

In the second setup, we investigate the compositional generalization capabilities of LLMs through the scope of dataset cartography and subset diversity. We observed that naïvely applying dataset cartography without a diversity incentive results in subpar performance while fine-tuning decoder-only LLMs compared to full fine-tuning performance. When we investigate the reason for this mediocre performance, we observe that every dataset cartography subset focuses on certain subsets while neglecting some of the remaining subspaces in the embedding space (see Figure 1.4). Thus, we propose a new subset selection method called Diverse CARTography

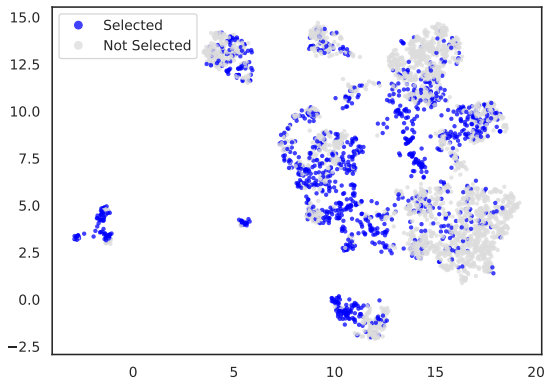

Figure 1.2: Dataset Cartography

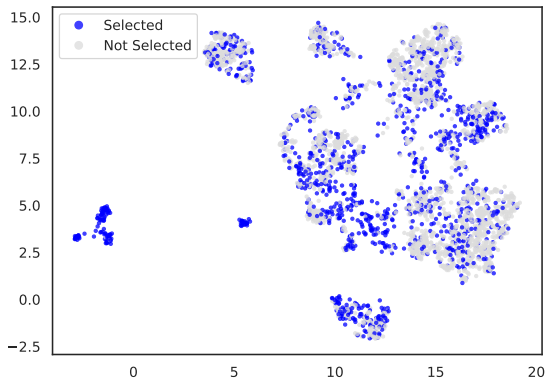

Figure 1.3: DiCart

Figure 1.4: UMAP [McInnes et al., 2018] plots for some subset selection methods on the training set of ATIS template split where TinyLlama 1.1B training dynamics and features are used. DiCart promotes diversity among selected examples (Subfigure 1.3) while prioritizing examples in dataset cartography (Subfigure 1.2).

(DiCart) for subset fine-tuning based on dataset cartography [Swayamdipta et al., 2020] and determinantal point process (DPP) [Kulesza et al., 2012]. We demonstrate that while DPP and especially dataset cartography remains limited in generalization performance, DiCart overall outperforms both baselines and full training subset.

In summary, our thesis makes the following key contributions: First, we introduce the novel use of dataset cartography as both a curriculum learning criterion and a sample selection strategy for enhancing compositional generalization. By leveraging dataset cartography, we enable models to deal effectively with the complexities of compositional tasks. Second, we thoroughly investigate the effectiveness of various confidence measures for sequences in extracting dataset cartography within the compositional generalization setting. This analysis provides insights into quantifying the difficulty of sequences and leads to the development of robust training strategies. Third, through extensive analyses, we demonstrate the significant impact of leveraging training dynamics through dataset cartography on the compositional generalization capabilities of Transformer models. Our approach yields significant improvements of up to 10% on challenging CFQ and COGS datasets, highlighting

the effectiveness of our proposed method. Fourth, we propose a new data selection mechanism for LLMs that consistently outperforms or performs on par with baselines and full training in four compositional generalization tasks. Fifth, other baselines rarely outperform DiCart in certain settings, but they also critically fail in other setups while DiCart is robust across all settings. Lastly, DiCart incorporates diversity over dataset cartography without any additional process by using the fine-tuned LLM used to record training dynamics.

Chapter 2

RELATED WORK

[Swayamdipta et al., 2020] use **training dynamics** to create data maps that categorize the dataset into three groups: easy-to-learn, hard-to-learn, and ambiguous. In a similar vein, [Toneva et al., 2019] employ training dynamics for dataset categorization, specifically in classification tasks, by identifying misclassified or forgotten instances. On the contrary, the adversarial filtering algorithm proposed by [Le Bras et al., 2020] ranks instances based on predictability, suggesting the removal of easy-to-learn examples. However, our research presents contrasting findings. Our experimental analyses show that combining the easy-to-learn category with other categories can improve the generalization performance. In another recent study, [Wang et al., 2022] explore the relationship between the generalization performance and the training dynamics in an active learning setting. Their approach revolves around the adaptive selection of samples for labelling to obtain comparable or better performance with less training data. Notably, they discovered a robust correlation between the convergence speed of training and the resulting generalization performance. By leveraging this connection, they propose a strategy to enhance overall generalization performance.

Contemporary research on **compositional generalization** focuses on two main aspects: proposing new datasets to explore model generalization capabilities and introducing novel techniques to address the compositionality problem.

There are multimodal compositional generalization **datasets** such as the CLEVR-CoGenT split from the CLEVR [Johnson et al., 2017] and CLOSURE [de Vries et al., 2019] that measure compositional visual reasoning with synthetic images including simple 3D shapes. However, we mainly focus on text-only compositional generalization datasets. As one of the early sample datasets, SCAN [Lake and Baroni, 2017] simulates a navigation task, and measures generalization to longer splits or

systematic generalization with new verbs in different splits. [Kim and Linzen, 2020] proposed COGS for semantic parsing, where each data example is an English sentence paired with its logical form. [Keysers et al., 2020] define a mathematically rigorous way to create compositional datasets and create CFQ dataset which is a semantic parsing task of generating SPARQL queries from natural language questions. The data curation process in CFQ is called *maximum compound divergence* and the process maximizes the compound divergence between training and test sets while minimizing the atom divergence. GeoQuery [Zelle and Mooney, 1996] is another semantic parsing task, where the source is an English question about US geography and the target is its corresponding Prolog program. Three types of compositional splits are curated from GeoQuery: (1) TMCD split following the maximum compound divergence in CFQ dataset, (2) Template split where the program templates in training and test set are exclusive, (3) Length split where the test examples are longer than training examples. SMCalFlow [Andreas et al., 2020] comprises task-oriented English dialogues related to places, people, events, and weather paired with their dataflow programs. While this dataset is not inherently compositional, [Yin et al., 2021] propose a subset of SMCalFlow called SMCalFlow-CS having single-turn dialogues comprising two distinct domains, calendar event creation and organization structure. They curate multiple cross-domain (C) and single-domain (S) test sets for each domain, where the cross-domain set only consists of examples in which both domains are visible. For instance, if the training set of a cross-domain split includes 32 cross-domain examples, the split is dubbed 32-C. [Meron, 2022] further simplifies the annotations of SMCalFlow dataflow programs, resulting in shorter and more human-readable programs. Overnight [Wang et al., 2015] is a dataset composed of natural language and synthetic strings matched with their logical forms.

In terms of **novel techniques**, researchers propose various approaches for compositional generalization. These include creating **novel architectures** to solve the compositionality problem. [Andreas et al., 2016] propose modular and jointly-trained neural module networks (NMNs) that are dynamically composed based on the question in visual question answering. [Hudson and Manning, 2018] present Compositional Attention Networks, a fully differentiable neural network that har-

nesses dynamic memory, attention, and composition. Researchers also **modify existing architectures** for better generalization. [Russin et al., 2020] factorize alignment and translation by using a novel syntactic attention mechanism and [Akyurek and Andreas, 2021] improve generalization of neural decoders by incorporating lexicon learning to separate lexical factors from syntactical factors. Several works concentrate on utilizing different learning paradigms such as meta-learning [Lake, 2019, Lake and Baroni, 2023], pre-training [Furrer et al., 2020], multimodal learning [Bugliarello and Elliott, 2021, Yagcioglu et al., 2024], or data augmentation [Andreas, 2020, Qiu et al., 2022]. With the rise of large language models (LLMs), choosing in-context examples for better compositional generalization with LLMs [Levy et al., 2023, An et al., 2023] is another open research problem. In a similar work to ours, [Gupta et al., 2022] show that structurally diverse training in sample-efficient setups often speeds up and improves the generalization performance. While [Gupta et al., 2022] compare sampling techniques within a fixed set of examples, we compare training on training set subsets with the full training set. In the compositional generalization literature, only a few studies investigated the impact of training dynamics on generalization performance. For instance, studies by both [Liu et al., 2020] and [Chen et al., 2020] propose curriculum learning schemes to help models learn accurate execution traces for lengthy training samples. They divide the samples into partitions based on the length and train the models sequentially, starting with the shortest examples. In contrast, our work takes a different approach by utilizing training dynamics to create data maps and leveraging them for compositional generalization. This is achieved either through subset selection or curriculum criteria.

While general **data selection** techniques include core-set selection [Wei et al., 2013], forgetting [Toneva et al., 2019], and adversarial filtering [Le Bras et al., 2020], data selection techniques in LLM-era NLP can be categorized into five categories based on their learning stages – pre-training, instruction-tuning, alignment, in-context learning, and task-specific fine-tuning [Albalak et al., 2024]. These samplers select examples based on their quality, coverage, complexity, or combinations of these attributes.

For the pre-training stage, data curation methods such as semantic deduplication [Abbas et al., 2023], coverage sampling [Longpre et al., 2024], their combination [Tirumala et al., 2024], perplexity filtering [Wenzek et al., 2020, Muennighoff et al., 2024], and LLM-based filtering methods [Sachdeva et al., 2024] are used. In the task-specific fine-tuning stage, similar to our work, $\mathbb{D}^2$ pruning utilizes message passing over a dataset graph for a diverse and difficult subset selection [Maharana et al., 2023]. [Azeemi et al., 2023] collect cross-entropy loss of training examples during training to train a new model with top-k hard-to-learn examples.

Compositional generalization-oriented selection methods generally focus on in-context learning as recent LLMs display strong few-shot capabilities while highly sensitive to the presented examples and their ordering [Kumar and Talukdar, 2021]. The few-shot example selection methods can be classified into two categories: (1) learning-based methods and (2) learning-free methods. EPR is a learning-based method that uses an unsupervised retriever in conjunction with a scoring LM to curate a training set to train a dense retriever with a contrastive learning objective [Rubin et al., 2022]. [Ye et al., 2023] propose CEIL, which further incorporates DPP into their pipeline to sample a diverse set of in-context examples. [Drozdov et al., 2022] propose a novel prompting technique called *dynamic least-to-most prompting* that outperforms the popular chain-of-thought prompting strategy [Wei et al., 2022] in compositional tasks. However, this method also includes dynamic exemplar selection by carefully curated heuristics. [An et al., 2023] present that selecting simple examples similar to the test example while having a diverse pool of in-context exemplars results in the best in-context compositional generalization. [Gupta et al., 2023] propose a learning-free method, focusing on the coverage aspect of in-context examples.

While compositional generalization-oriented selection methods focus on in-context learning, several selection techniques for fine-tuning also exist. [Oren et al., 2021] propose a structurally diverse sampling method to sample synthetic examples that are used to train a model before fine-tuning the model on a small set of annotated examples. Similarly, [Bogin et al., 2022] and [Gupta et al., 2022] argue for structurally diverse sampling methods for efficient training, but they differ in how to measure the

structural diversity. Moreover, they compare sampling techniques within a fixed set of examples, whereas we compare with the full training set with a more ambitious aim. [İnce et al., 2023] propose the most similar approach to ours, where they first extend dataset cartography to generative modelling and then train a new model with the hard-to-learn examples to improve compositional generalization of models significantly.

Chapter 3

BACKGROUND

3.1 Compositional Generalization

[Partee et al., 1995] defines the principle of compositionality as *"The meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined."*. While this principle is thought to be a valid statement with support, there is no consensus on the exact implications of this definition on practical natural language processing. [Hupkes et al., 2020] aim to alleviate this variance by deeply investigating the linguistics and philosophy literature for compositionality and constructing different compositionality tests for neural models based on their findings. The authors recognize five aspects of compositional generalization: systematicity, productivity, substitutivity, localism, and overgeneralization.

3.1.1 Systematicity

Systematicity is the ability to recombine seen parts and rules to generate new combinations and is a frequently researched aspect of compositionality. Moreover, the postulated absence of systematicity in neural models is one of the main criticisms directed towards connectionist architectures. An example of systematic ability is as follows: someone who understands the meanings of `dax`, `jump`, and `jump twice` should know what `dax twice` means [Lake and Baroni, 2017].

3.1.2 Productivity

The productive aspect of compositionality expresses the open-endedness of natural language. Similar to systematicity, productivity also involves the combination of expressions. However, productivity focuses more on the length side as a natural language expression can include possibly infinite expressions. An example of pro-

ductivity is shown in the following sentence with a potentially infinite tail: A teacher gives lectures in a school, near a lake, by the house, ...

3.1.3 Substituvity

The principle of substitutivity states that if a part of an expression is altered with another synonymous part, the meaning of the expression should not be affected. The substitutivity principle is not as incontestable as there are exceptions to it [Geach, 1965]. Nonetheless, it is a worthy aspect of compositionality to examine.

3.1.4 Localism

Localism is related to the level of compositionality, meaning how global or local the parts of a whole are interpreted. *Strong* (or very local) operations mainly involve only the meanings of its intermediate parts and the local structure, while the parts in *weak* (or global) compositionality can have different meanings depending on the global structure they are part of. Therefore, localism is a more controversial aspect of compositionality.

3.1.5 Overgeneralization

The last aspect is overgeneralization, which focuses on the non-compositional aspects of the language. In Turkish, the non-continuant fortis consonants at the end of words go under *lenition* after they take a suffix starting with a vowel (e.g., kağıt + ı → kağıdı, dolap + a → dolaba). While this transformation generally holds, a few exceptions exist (e.g., sepet + i → sepeti). [Hupkes et al., 2020] hypothesize that if a model overgeneralizes while applying rules where "sepet" + "i" becomes **sepedi** instead of the correct spelling **sepeti**, the model displays compositional awareness.

3.2 Dataset Cartography

[Swayamdipta et al., 2020] propose a visualization tool named **data maps** with two dimensions, **confidence** and **variability**, which characterizes the informativeness of training instances of a dataset concerning a model. Confidence is calculated as

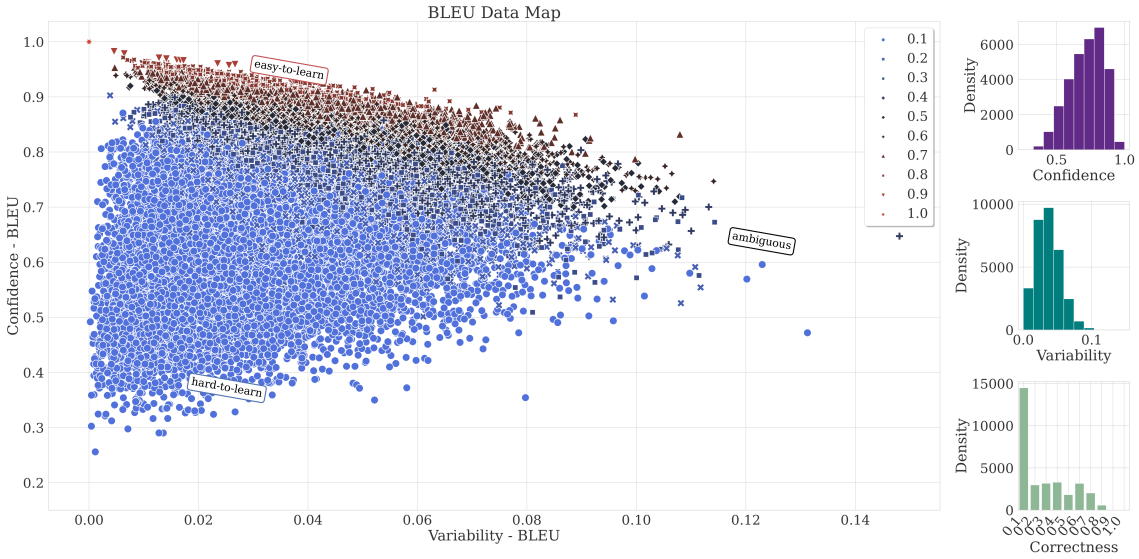

Figure 3.1: **Data map of CFQ train set** for the Transformer model based on BLEU measure (converge epoch 20). The x-axis shows the **variability** and the y-axis the **confidence**. The colours and shapes indicate the **correctness**.

the mean probability of the true label across epochs, whereas variability corresponds to the spread of confidence across epochs, using the standard deviation. Therefore, confidence ($\hat{\mu}_i$) and variability ($\hat{\sigma}_i$) is denoted as follows:

$$\hat{\mu}_i = \frac{1}{E} \sum_{e=1}^{E} p_{\theta^{(e)}}(y_i^* | \mathbf{x}_i) \tag{3.1}$$

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{e=1}^{E} (p_{\theta^{(e)}}(y_i^* | \mathbf{x}_i) - \hat{\mu}_i)^2}{E}} \tag{3.2}$$

where i denotes the instance, E represents the total number of epochs, $\mathbf{x}_i$ is the input sequence, y_i^* is the true label, $\theta^{(e)}$ corresponds to the set of model parameters at epoch e.

Data maps reveal three distinct regions: **ambiguous** instances (high variability), **easy-to-learn** instances (high confidence, low variability), and **hard-to-learn** instances (low confidence, low variability). [Swayamdipta et al., 2020] experimented on multiple NLI datasets such as SNLI [Bowman et al., 2015] with pretrained models and reported three main findings: (i) Ambiguous regions contribute the most

towards OOD generalization, (ii) Easy-to-learn regions play an important role in model optimization, (iii) Hard-to-learn regions often correspond to labelling errors.

3.3 Determinantal Point Processes (DPPs)

One of the prevalent ways of diverse subset selection is using DPPs [Kulesza et al., 2012]. Formally, a DPP $\mathcal{P}$ forms a probabilistic measure over all $2^{|\mathcal{D}|}$ subsets of a discrete set $\mathcal{D}$. DPP calculates a $|\mathcal{D}| \times |\mathcal{D}|$ positive semi-definite kernel matrix $\mathbf{L}$ by using the feature vector $\mathbf{v}$ associated with each element. Specifically, $\mathbf{L}_{ij}$ is calculated as $\mathbf{k}(\mathbf{v}_i, \mathbf{v}_j)$ where $\mathbf{k}(\cdot, \cdot)$ is a kernel function and i and j denote specific elements of $\mathcal{S}$. Then, the probability of selecting the subset $\mathcal{S} \subseteq \mathcal{D}$ is defined as

$$\mathcal{P}(\mathcal{S}) = \frac{\det(\mathbf{L}_{\mathcal{S}})}{\det(\mathbf{L} + \mathbf{I})}. \tag{3.3}$$

$\mathbf{L}_{\mathcal{S}}$ matrix in the Equation 3.3 is equivalent to $[\mathbf{L}_{ij}]_{i,j \in \mathcal{S}}$ and $\mathbf{I}$ is the identity matrix with the same dimensions as $\mathbf{L}$. We can express $\mathbf{k}(\mathbf{v}_i, \mathbf{v}_j)$ as $\phi(\mathbf{v}_i)^T \phi(\mathbf{v}_j)$ due to the kernel trick where the $\phi(\cdot)$ represents a reproducing kernel map [Schölkopf and Smola, 2002]. When the magnitude of the feature vector $\mathbf{v}_i$ grows, the probability of subsets including i increases. Additionally, $\mathbf{v}_i$ and $\mathbf{v}_j$ becoming more similar decreases the probability of subsets simultaneously including i and j. To decrease the exponential complexity of DPP caused by the number of possible subsets, many DPP inference methods employ techniques such as sampling and MAP inference [Kulesza et al., 2012, Chen et al., 2018].

Chapter 4

DATA MAPS FOR GENERATIVE TASKS

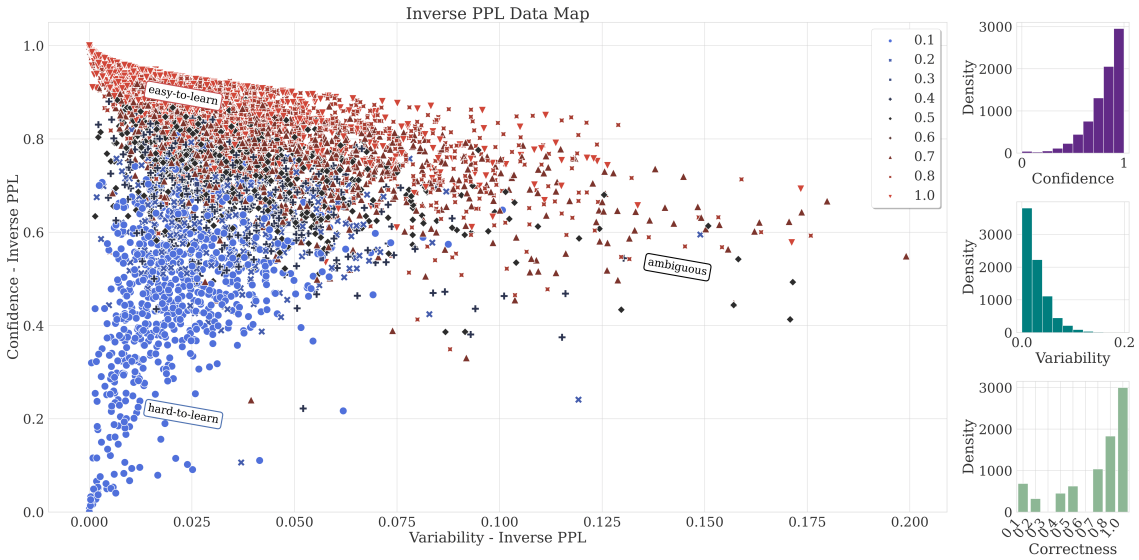

Figure 4.1: **Data map of COGS train set** for the Transformer model based on Inv PPL measure (converge epoch 10). The x-axis shows the **variability** and the y-axis the **confidence**. The colours and shapes indicate the **correctness**.

We propose various measures to adapt the dataset cartography to generative modelling to improve compositional generalization in models trained from scratch. We expand the dataset cartography to generative tasks by proposing to use the inverse perplexity (Inv PPL) metric, outperforming any baselines in the literature. After this expansion, we enhance a model's learning process and promote better generalization by identifying and focusing on informative and helpful examples. Additionally, we demonstrate that using dataset cartography for curriculum learning can improve the compositionality of models as well, adding another layer of novelty to our work. To test our hypotheses, we conduct extensive experiments using three compositional generalization datasets (COGS, CFQ, SMCalFlow-CS) and two

neural architectures (Transformers, Bi-LSTM). Our results show that dataset cartography improves the compositional generalization of models trained from scratch in subset selection and curriculum learning setups.

4.1 Approach

The notions of confidence and variability in Eq. (3.1) and (3.2) are defined considering classification-based tasks, and thus not directly applicable to seq2seq models. Extending dataset cartography to machine translation, a generative task, [Bhatnagar et al., 2022] propose the CHIA measure by following the intuition that an output sequence consists of a series of predictions. Instead of using the exact match, they take the arithmetic mean of gold token predictions over the sequence, defined as:

$$\hat{\mu}_i = \frac{1}{ET} \sum_{e=1}^{E} \sum_{t=1}^{T} p_{\theta^e} \left(y_{it}^* \mid \mathbf{x}_i, (y_{i\tau}^*)_{\tau=1}^{t-1} \right) \tag{4.1}$$

where y_{it}^* corresponds to the i-th token of the groundtruth output sequence $\mathbf{y}_i$ of length T.

The variability of CHIA is denoted as:

$$v_i = \sqrt{\frac{\sum_{e=1}^{E} \left(\frac{1}{T} \sum_{t=1}^{T} p_{\theta^e} \left(y_{it}^* \mid \mathbf{x}_i, (y_{i\tau}^*)_{\tau=1}^{t-1} \right) - \hat{\mu}_i \right)^2}{E}} \tag{4.2}$$

Here, it is important to note that [Bhatnagar et al., 2022] do not use data maps to select a subset but instead use N-way translation corpora to choose instances that are most informative on all ways to select instances to annotate for low-resource languages. They showed that choosing instances based on a single-way translation decreases performance significantly, suggesting CHIA measure might not be the best choice for our experimental setting.

Similar to the CHIA score, we also consider inverse perplexity (Inv PPL) for the reason that high perplexity is an undesirable property. It is defined as the geometric mean of gold token predictions over the sequence, as given below:

$$\hat{\mu}_i = \frac{1}{E} \sum_{e=1}^{E} \prod_{t=1}^{T} \sqrt[T]{p_{\theta^e} \left(y_{it}^* \mid \mathbf{x}_i, (y_{i\tau}^*)_{\tau=1}^{t-1} \right)} \tag{4.3}$$

Similarly, the variability of Inv PPL is given below:

$$v_i = \sqrt{\frac{\sum_{e=1}^{E} \left(\prod_{t=1}^{T} \sqrt[T]{p_{\theta^e} \left(y_{it}^* \mid \mathbf{x}_i, (y_{i\tau}^*)_{\tau=1}^{t-1} \right)} - \hat{\mu}_i \right)^2}{E}} \tag{4.4}$$

Equations for variability calculations, when CHIA, inverse perplexity, or BLEU measures are used, are shown in equations 4.2, 4.4, 4.6 respectively. In the dataset cartography equations, i denotes the instance, E represents the total number of epochs, $\mathbf{x}_i$ is the input sequence, $\theta^{(e)}$ corresponds to the set of model parameters at epoch e. Additionally, $\hat{\mu}_i$ denotes confidence for the instance i, y_{it}^* corresponds to the t-th token of the ground truth output sequence $\mathbf{y}_i$ of length T and $(y_{i\tau}^*)_{\tau=1}^{t-1}$ denotes the ground truth output sequence until the t-th token. Lastly, $\hat{\mathbf{y}}_\mathbf{i}^{(e)}$ refers to the predicted sequence generated by the model parameters at epoch e.

The geometric mean is much closer to the lowest probability in the sequence compared to the arithmetic mean used in CHIA, making inverse perplexity a more discriminative measure. Additionally, perplexity has potentially preferable information theoretical properties.

Additionally, we define a third measure based on BLEU [Papineni et al., 2002]. In particular, BLEU measures the n-gram overlap between generated output and the ground truth, and we use the arithmetic mean of the BLEU score across epochs as a means to measure the confidence and variability as follows:

$$\hat{\mu}_i = \frac{1}{E} \sum_{e=1}^{E} \text{BLEU}(\hat{\mathbf{y}}_\mathbf{i}^{(e)}, \mathbf{y_i}) \tag{4.5}$$

$$v_i = \sqrt{\frac{\sum_{e=1}^{E} \left(\text{BLEU}(\hat{\mathbf{y}}_\mathbf{i}^{(\mathbf{e})}, \mathbf{y}_i) - \hat{\mu}_i \right)^2}{E}} \tag{4.6}$$

where $\hat{\mathbf{y}}_\mathbf{i}^{(e)}$ refers to the predicted sequence generated by the model parameters at epoch e, and $\mathbf{y_i}$ denotes the ground truth sequence, respectively. A particular shortcoming of using BLEU is its computational and temporal expense due to decoding compared to CHIA and Inv PPL as these methods do not require decoding.

The main motivation behind utilizing dataset cartography lies in selecting a subset of the training set and training the model on these subsets instead of the entire dataset. The selection process involves two key considerations: (1) Choosing the measure used to rank the examples, and (2) Determining the aspect of the measure scores for ranking (e.g., ambiguity). There are more hyperparameters such as subset ratio and subset combinations, until which epoch to take training dynamics into account (referred to as **convergence epoch**), from starting with which epoch training dynamics are considered (referred to as **min epoch**). Specifically, to identify the convergence epochs, we qualitatively examine loss convergence and generated data maps. Unlike [Swayamdipta et al., 2020], where authors utilize pre-trained models and consider training dynamics from the start of fine-tuning, our experimental setup involves randomly initialized models. Hence, considering training dynamics in the initial epochs while the model is unstable can result in noisy training dynamics [Swayamdipta et al., 2020], and can introduce selection biases based on data ordering. To simplify the process, we set the minimum epoch to 3 across our training setups in setup 1.

4.2 Experiments

4.2.1 Baselines

We benchmark with several baselines to demonstrate the enhancement in generalization performance through the selection of smaller, specifically chosen subsets. The most rudimentary of these baselines involves the selection of a random subset identical in size to the specifically chosen subset, along with the utilization of the entire original dataset for comparison, i.e. 100% of the original dataset. In the context of curriculum learning settings, we deem it necessary to establish another benchmark wherein no particular curriculum is employed. This serves as a baseline, facilitating the process of benchmarking for comparative purposes.

4.2.2 Datasets

We conduct our experiments on three compositional generalization datasets, CFQ [Keysers et al., 2020], COGS [Kim and Linzen, 2020], SMCalFlow-CS [Andreas et al., 2020, Yin et al., 2021] Simple [Meron, 2022, SMCS] datasets. CFQ and SMCS dataset has multiple splits. For CFQ, we utilize the MCD1 split. For SMCS, we use 16-C and 32-C compositional splits, where the split numbers refer to the cross-domain example leak count into the training set. One challenge commonly encountered with compositional generalization datasets in the literature is the absence of validation splits. To ensure a fair comparison, we train all of our models with specified step counts, following the approach of [Csordás et al., 2021].

To provide a better understanding of the datasets, let us consider specific examples from each. CFQ, being a synthetic text-to-SQL dataset, involves input samples such as "*Did a male film director produce and edit M1?*" with the corresponding target query being `SELECT count(*) WHERE {?x0 ns:film.producer.film M1 . ?x0 ns:film.editor.film M1 . ?x0 ns:people.person.gender m_05zppz}`. In the case of COGS, which is a synthetic semantic parsing task, an input sample could be "*A frog hopped*" and the corresponding target logical form is `frog(x1) AND hop.agent(x2, x1)`. For the natural semantic parsing dataset SMCS, an input sample is "*iam meet with smith , john and rodney*", and its output is `CreateEvent( AND( with_attendee( rodney ) , with_attendee( smith ) , with_attendee( john ) ) )`.

4.2.3 Experimental Setup

In our experiments, we employ the vanilla Transformer model [Vaswani et al., 2017]. Recent studies have highlighted that the generalization capabilities of pre-trained Transformer models can be overestimated due to uncontrolled lexical exposure [Kim et al., 2022, An et al., 2023]. We adopted the publicly available PyTorch codebase provided by [Csordás et al., 2021] to implement our model. Each experiment is executed on a single Tesla T4 GPU. We employ a whitespace tokenizer for all datasets, considering that the targets in these datasets are not expressed in natural language.

Dataset	**#train**	**#test**	**Voc. size**	**Train len.**	**Test len.**
CFQ	95743	11968	181	29 / 95	30 / 103
COGS	24155	21000	871	22 / 153	61 / 480
SMCS 16-C	25410	663	10738	107 / 103	30 / 59
SMCS 32-C	25426	662	10738	107 / 103	30 / 59

Table 4.1: Dataset statistics showing sample counts, vocabulary size as the combined input and output vocabularies, and train and test length denoting the max. input/output length in the train and test set, respectively.

We also experiment with Bi-LSTM with attention [Bahdanau et al., 2015] on the COGS dataset.

In a similar way to [Swayamdipta et al., 2020], we show the data maps generated for CFQ based on BLEU and COGS based on Inv PPL in Figure 3.1 and Figure 4.1, respectively. For better visualizations, we only plot randomly sampled 33% of the training set. Considering a larger number of training epochs compared to [Swayamdipta et al., 2020], we divide the correctness scores into 10 bins for better granularity and informative visualizations. As we discussed earlier, we use three distinct confidence measures, inverse perplexity (Inv PPL), CHIA [Bhatnagar et al., 2022], and BLEU [Papineni et al., 2002]. The omitted data maps are given in Appendix A.3.

We explore two different setups to assess the effectiveness of leveraging data maps in improving compositional generalization. In the first setup, we utilize subsets comprising 33% of the original datasets. These subsets are categorized as *hard-to-learn*, *ambiguous*, and *easy-to-learn* based on the extracted maps. For the second setup, we train models using subsets sized at 50% of the original datasets. Along with the *hard-to-learn*, *ambiguous*, and *easy-to-learn* subsets, we construct combined subsets that are also half the size of the original dataset by merging two 33% subsets selected based on the same confidence measure. Specifically, we select 33% of the examples from the more informative subset and allocate the remaining 17% from

the other subset, following [Swayamdipta et al., 2020]. As will be discussed in the next section, our findings demonstrate that *hard-to-learn* samples have a more pronounced impact on model performance compared to *ambiguous* and *easy-to-learn* samples, and thus we consider them as more informative. When combining *ambiguous* and *easy-to-learn* samples, we consider including a greater number of *ambiguous* samples than *easy-to-learn* samples. If the union of these subsets is smaller than half of the whole training data, we randomly add samples to reach the 50% dataset size. Furthermore, we address the out-of-vocabulary (OOV) problem during subset selection by incorporating training samples from the entire dataset if they increase the vocabulary size. On the contrary, we remove the least informative samples that do not reduce the vocabulary size, ensuring consistent subset sizes throughout the experiments. The statistics about the subsets obtained from the data maps are provided in Appendix 4.2.7.

In addition to our subset selection experiments, we explore the potential of leveraging dataset cartography as a criterion for curriculum learning (CL). In particular, we adopt two CL approaches proposed by [Hacohen and Weinshall, 2019] and [Zhang et al., 2019]. We experiment with a fixed exponential pacing schedule using default hyperparameters in the former. We set the starting percentage to 4% and increase the scale to 1.9. On the other hand, the second CL method by [Zhang et al., 2019] involves sorting examples based on a given criterion and dividing them into 10 equal-sized bins, resulting in a 10-stage curriculum. Within each bin, the examples are further sorted based on their lengths, and then each sorted bin is divided into non-overlapping batches. We distribute these batches randomly during training to avoid potential selection bias. Since we train our models for a fixed number of steps, after completing $1/10^{th}$ of the training, we unlock the second bin in a similar fashion.

4.2.4 Impact of Selected Subsets

We conduct a thorough analysis to understand the effect of subset selection on the training process and how the selection process impacts the subsequent generalization abilities of the models. Our key findings are summarized in Table 4.2 for the

Table 4.2: Accuracy results for CFQ and COGS datasets. Models are trained on different 33% subsets of the train data compared to using the full dataset. The scores are averaged over 3 runs, where std. dev. is shown as a subscript. The best and second-best performing subsets are highlighted in bold and underlined, respectively. *Hard-to-learn* subset consistently performs better than the *random* subset, even outperforming 100% train set on the COGS dataset.

		CFQ			**COGS**		
		Inv PPL	**CHIA**	**BLEU**	**Inv PPL**	**CHIA**	**BLEU**
33% train	*easy-to-learn*	$12.19_{1.20}$	$12.42_{0.59}$	$9.88_{1.83}$	$0.00_{0.00}$	$0.06_{0.11}$	$0.04_{0.07}$
	ambiguous	$17.69_{0.47}$	$23.51_{0.86}$	$20.99_{1.91}$	$3.26_{5.61}$	$20.30_{3.58}$	$26.69_{4.17}$
	hard-to-learn	$\mathbf{36.55}_{0.55}$	$\mathbf{34.98}_{0.67}$	$\mathbf{34.71}_{1.12}$	$\mathbf{53.50}_{6.80}$	$\mathbf{45.41}_{12.5}$	$\mathbf{50.56}_{3.07}$
	random		$\underline{34.02}_{1.09}$			$18.66_{6.72}$	
100% training			$38.71_{1.01}$			$42.54_{7.62}$	

subsets comprising 33% of the original datasets. Our experimental results show that training models on *hard-to-learn* samples consistently yields superior generalization performance compared to training on ambiguous samples. Notably, the performance of *hard-to-learn* subsets surpasses that of *random* subsets overall, and for the COGS dataset, it even outperforms training on the entire training set. Training the models on *easy-to-learn* samples, on the other hand, leads to poor generalization performance. We also observe that Inverse Perplexity is a more effective measure than CHIA or BLEU for selecting samples based on their difficulty.

As we increase the subset size to 50% of the original dataset, our experimental results demonstrate significant improvements compared to full dataset training, as shown in Table 4.3. In the CFQ dataset, the accuracy of the *hard-to-learn (Inv PPL)* subset exceeds that of the full training by over 4%. When considering the CHIA measure, both the *hard-to-learn* and *hard-to-learn+easy-to-learn* subsets outperform 100% training. However, when using the BLEU measure, only the *hard-to-learn+easy-to-learn* subset surpasses the 100% training performance. Although the subset combinations show promising results with the CHIA and BLEU measures,

Table 4.3: Accuracy results for CFQ and COGS datasets. Models are trained on different 50% subsets of the train data compared to using the full dataset. The best and second-best performing subsets are highlighted in bold and underlined, respectively. It is worth mentioning that solely training on *hard-to-learn* samples or combining them with *easy-to-learn* samples outperforms using 100% training data.

		CFQ			**COGS**		
		Inv PPL	**CHIA**	**BLEU**	**Inv PPL**	**CHIA**	**BLEU**
50% train	*easy-to-learn*	21.13	20.96	17.04	0.000	0.000	0.695
	ambiguous	23.03	28.80	24.31	0.047	36.09	35.14
	hard-to-learn	**42.45**	<u>40.13</u>	37.45	**47.48**	**42.40**	**45.20**
	ambiguous + easy-to-learn	18.52	26.18	20.77	0.048	18.33	25.69
	hard-to-learn + ambiguous	36.54	36.87	37.13	<u>41.13</u>	35.08	<u>41.16</u>
	hard-to-learn + easy-to-learn	35.91	**41.29**	**39.29**	40.82	<u>37.94</u>	40.96
	random		35.16			30.24	
100% training			37.71			36.80	

they are still outperformed by the *hard-to-learn (Inv PPL)* subset. In COGS, we observe even more substantial improvements in accuracy. Across all measures, the *hard-to-learn* subset demonstrates an accuracy increase of over 5%, with the *hard-to-learn (Inv PPL)* subset outperforming the 100% training by over 10% accuracy. Notably, selecting 50% of the *hard-to-learn* samples consistently outperforms the subset combinations for all measures. While combining subsets does yield performance improvements in certain measures, it also highlights the limited effectiveness of these measures in effectively separating the different classes of instances. This is evident as the *hard-to-learn (Inv PPL)* subset consistently outperforms the subset combinations in both the CFQ and COGS datasets.

4.2.5 Impact of Cartography-Based Curriculum Learning

We use dataset cartography to examine the impact of training dynamics on curriculum learning. Curriculum learning is a strategy that trains models on instances from

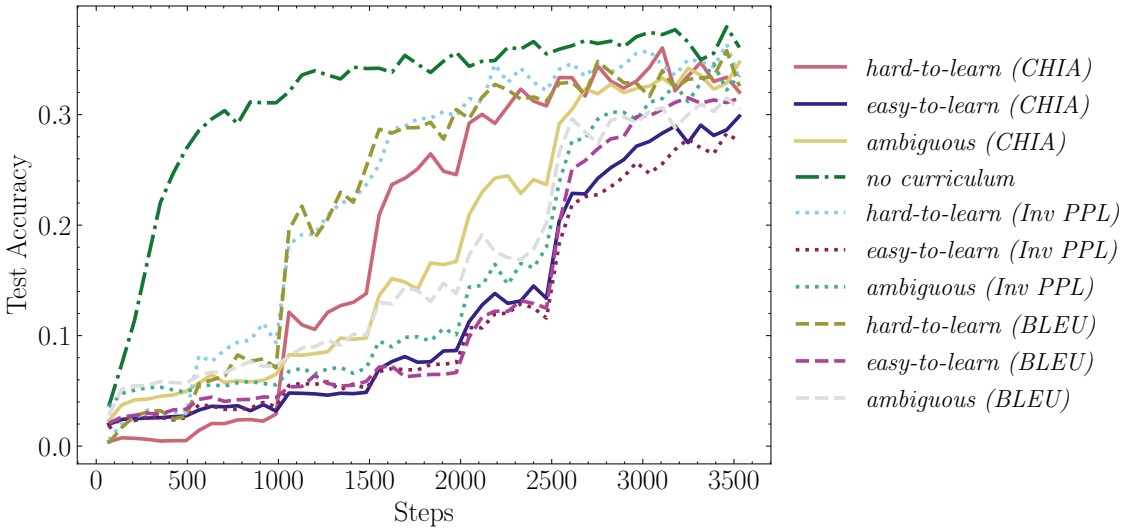

Figure 4.2: Accuracy plots on CFQ for the CL strategy by [Hacohen and Weinshall, 2019].

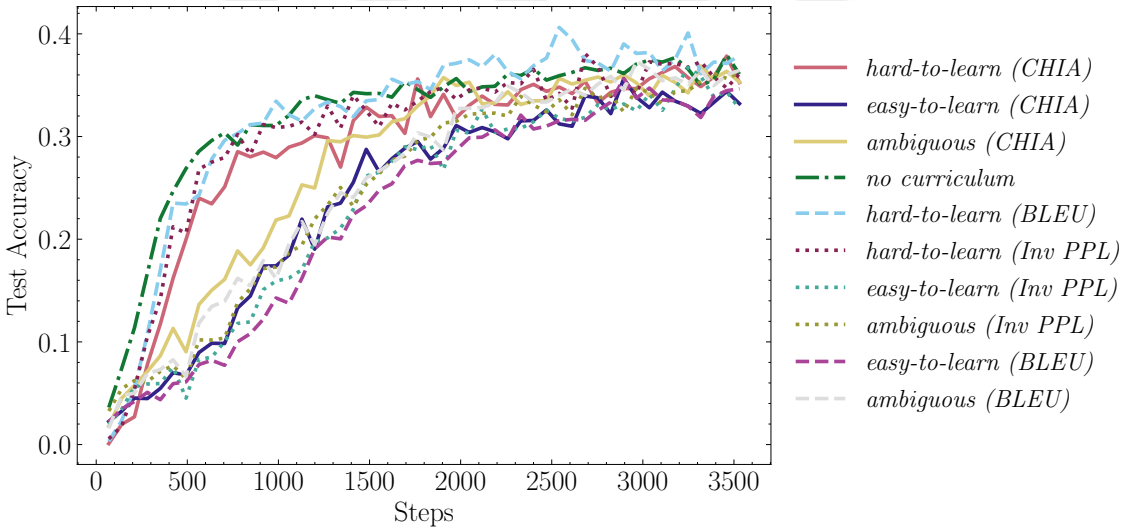

Figure 4.3: Accuracy plots on CFQ for the CL strategy by [Zhang et al., 2019]

easy to hard, based on the assumption that this order facilitates learning. However, we also explore the opposite strategy, which trains models on instances from hard to easy, and compare it with the conventional curriculum learning approach. This way, we can study how different training schedules affect the model performance.

Figure 4.2 depicts accuracy plots showing the performance of various CL strate-

gies based on [Hacohen and Weinshall, 2019] on the CFQ dataset. The figure legends indicate the ranking scheme and the employed confidence measure. For instance, *hard-to-learn (Inv PPL)* refers to the case where Inv PPL is being used as the confidence measure, and the inclusion of the *hard-to-learn* samples is prioritized within the curriculum. Our analysis reveals that no single curriculum consistently outperforms others on the CFQ dataset. Exponential pacing leads to stagnant performance in the final $2/7^{th}$ of the training process due to surpassing the training size percentages of 33% and 50%. Surprisingly, initiating training with *hard-to-learn* samples yields superior performance compared to *easy-to-learn* samples, contrary to common curriculum learning expectations. This aligns with our previous findings, emphasizing the benefits of starting with challenging examples for improved adaptation.

Figure 4.3 examines the impact of leveraging data maps within the CL strategy proposed by [Zhang et al., 2019] for compositional generalization. The *hard-to-learn (BLEU)* configuration outperforms the *no curriculum* strategy, albeit with no notable improvement in convergence speed. This outcome mirrors our observations using the CL framework developed by [Hacohen and Weinshall, 2019], where initiating training with harder samples leads to better performance. However, the *ambiguous* configurations perform similarly to *no curriculum*, while the *easy-to-learn* configurations yield worse results than the *no curriculum* approach.

In Figures 4.4 and 4.5, we gain deeper insights into the contributions of dataset cartography. Overall, *hard-to-learn (BLEU)* emerges as the most effective configuration in the plots. Surprisingly, *ambiguous (Inv PPL)* performs as the second-best configuration in Figure 4.5, while *hard-to-learn (Inv PPL)* holds this position in Figure 4.4. The *no curriculum* approach ranks third and fourth in these respective plots. Furthermore, the *easy-to-learn* configurations demonstrate the poorest final performance across both curriculum learning frameworks.

Analyzing the accuracy plots of curriculum learning, we observe that initiating training with easier examples and gradually progressing to more challenging instances does not lead to accelerated convergence or improved final model performance. On the other hand, the subset experiments presented in Tables 4.2 and 4.3 show that training models on hard-to-learn examples result in better model per-

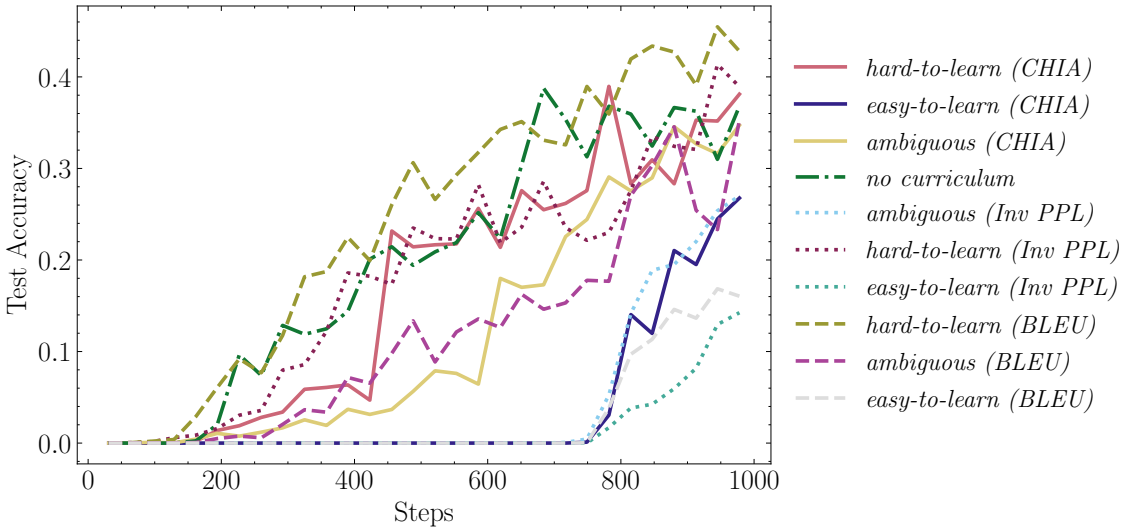

Figure 4.4: Accuracy plots on COGS for the CL strategy by [Hacohen and Weinshall, 2019].

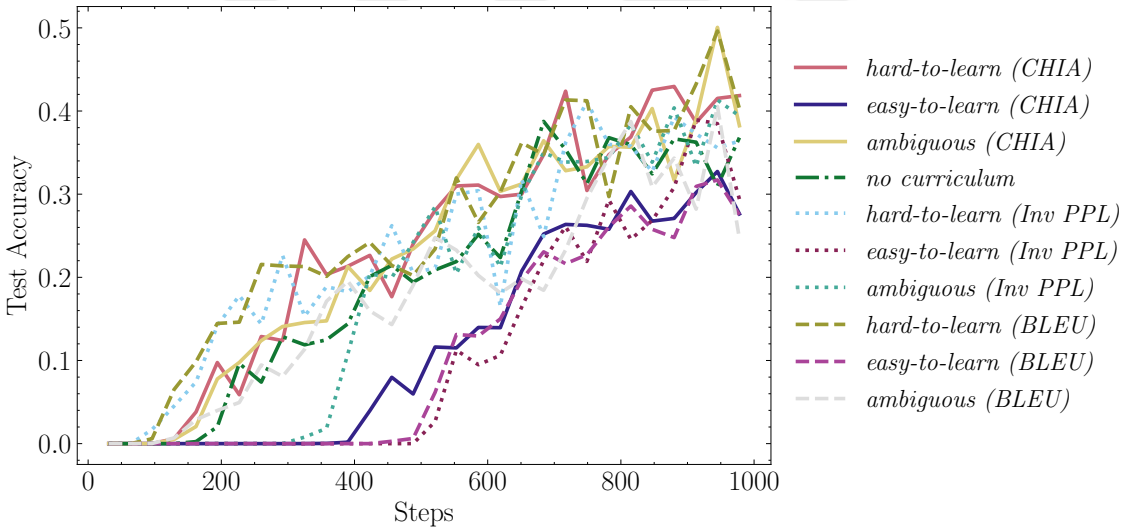

Figure 4.5: Accuracy plots on COGS for the CL strategy by [Zhang et al., 2019].

formance. Furthermore, the CL results highlight that starting the curriculum with *hard-to-learn* samples results in enhanced final performance. These findings, combined with the observation that the first unlocked examples are encountered more frequently during training, suggest that the superiority of *hard-to-learn* curricula over the *no curriculum* can be attributed to the increased exposure to challenging

instances throughout the training process.

To sum up, our experimental findings highlight the effectiveness of utilizing dataset cartography for training subset selection and curriculum learning in the context of compositional generalization. Our results consistently show that leveraging dataset cartography leads to improved generalization performance. While curriculum learning also contributes to performance enhancement, its impact appears to be smaller compared to the use of dataset cartography for subset selection.

4.2.6 Additional Experiments

	SMCS 16-C			**SMCS 32-C**		
	Inv PPL	**CHIA**	**BLEU**	**Inv PPL**	**CHIA**	**BLEU**
easy-to-learn	0.0	0.0	0.0	0.1	0.0	1.5
ambiguous	0.0	2.0	2.3	7.0	7.8	11.6
hard-to-learn	**4.5**	**4.5**	**6.8**	**16.8**	**17.5**	**15.9**
random	0.4	0.4	0.4	5.9	5.9	5.9
100% train		4.2			15.6	

Table 4.4: Accuracy results for the SMCS 16-C and 32-C splits. Models are trained on different 50% subsets of the train data instead of the full train set. The best-performing subset is given in bold. Training models only on *hard-to-learn* samples outperform using 100% train data.

While our primary experiments focus on training Transformers on the CFQ and COGS datasets, we conduct supplementary experiments with Bi-LSTM with attention model on the COGS task and Transformer model on two SMCS splits. This approach allows us to examine the transferability of our results from synthetic datasets and the Transformer to other architectures and natural datasets.

For the SMCS 16-C and 32-C splits, we see a similar trend compared to CFQ and COGS results (see Table 4.4 and 4.3). The performance of *hard-to-learn* subsets exceeds the original dataset performance consistently. Even if *hard-to-learn* examples consisted of manual annotation errors in [Swayamdipta et al., 2020], training

	Inv PPL	**CHIA**	**BLEU**
easy-to-learn	2.7	8.3	9.9
ambiguous	14.2	2.8	11.9
hard-to-learn	**16.6**	**15.7**	**20.2**
random		9.9	
100% train		13.7	

Table 4.5: Accuracy results for the COGS dataset. Models are trained on different 50% subsets of the train data instead of the full train set. The best-performing subset is given in bold. Training models solely on *hard-to-learn* samples outperform using 100% train data.

models on *hard-to-learn* subsets improve performance for the SMCS splits. However, this finding does not necessarily translate to training models on errors resulting in better performance. Rather, the contribution of *hard-to-learn* subsets significantly obscures any performance degradation that may occur from annotation errors. Similarly, *easy-to-learn* subsets perform the worst among all of the subsets. Although *hard-to-learn* subsets perform the best among the other subsets, the ranking between metrics is more fluid compared to the CFQ and COGS results. While *hard-to-learn (Inv PPL)* outperform other subsets persistently in the CFQ and COGS results (Table 4.3), *hard-to-learn (BLEU)* and *hard-to-learn (CHIA)* are the best-performing subsets in SMCS 16-C and 32-C subsets respectively.

Table 4.5 shows the Bi-LSTM with attention performance on the COGS dataset. Surprisingly, the Bi-LSTM performance is much worse than the Transformer performance. Nonetheless, these results are consistent with the results in the original COGS dataset [Kim and Linzen, 2020].

Similar to the SMCS experiments, the Bi-LSTM experiments support our primary findings. Training models on *hard-to-learn* subsets continuously outperform training on the full dataset. Compared to the Transformer performance (Table 4.3), the maximum absolute performance increase between *hard-to-learn* subsets and full

		Length	**Word rarity**
	random	13.56 / 27.78	3.97 / 3.47
Inv PPL	*easy-to-learn*	11.86 / 24.55	3.99 / 3.41
	ambiguous	11.51 / 23.64	4.06 / 3.51
	hard-to-learn	15.82 / 32.13	3.95 / 3.54
CHIA	*easy-to-learn*	10.91 / 22.34	4.02 / 3.43
	ambiguous	13.07 / 27.86	3.94 / 3.50
	hard-to-learn	16.40 / 33.94	3.93 / 3.53
BLEU	*easy-to-learn*	11.41 / 23.18	4.02 / 3.44
	ambiguous	12.78 / 25.11	3.96 / 3.47
	hard-to-learn	16.23 / 33.49	3.94 / 3.52

Table 4.6: Statistics about the subsets of the CFQ dataset on 33% selected instances based on Inv PPL, CHIA and BLEU measures. We report average input/output length and word rarity. Statistics are averaged over 3 runs.

training decreases by 4%. However, the maximum relative performance increase between *hard-to-learn* subsets and full training increases, showing that dataset cartography improves generalization performance in architectures other than Transformer.

4.2.7 Subsets Obtained from Data Maps

We examine four key statistics to gain insights into the nature of the subsets created through data cartography: (1) input length, (2) output length, (3) input word rarity, and (4) output word rarity. Word rarity is calculated as the sum of negative log word frequencies normalized with sentence length, as shown in Equation 4.7. In this equation, T is the sequence length, y^*_{it} is the t^{th} gold token for sequence i, and $f(y^*_{it})$ denotes frequency of gold token y^*_{it}. Table 4.6 presents these statistics for the CFQ dataset, and reveals interesting patterns. Among these different subsets, the *hard-to-learn* subsets show longer input and output lengths compared to all other

		Length	**Word rarity**
	random	7.47 / 43.52	4.54 / 3.34
Inv PPL	*easy-to-learn*	6.59 / 34.22	4.25 / 3.24
	ambiguous	7.16 / 42.23	4.82 / 3.41
	hard-to-learn	8.83 / 56.62	4.87 / 3.47
CHIA	*easy-to-learn*	6.61 / 33.78	4.24 / 3.23
	ambiguous	7.81 / 48.53	4.90 / 3.46
	hard-to-learn	8.83 / 56.87	4.87 / 3.48
BLEU	*easy-to-learn*	6.27 / 32.24	4.33 / 3.27
	ambiguous	8.67 / 54.92	4.78 / 3.43
	hard-to-learn	9.08 / 58.20	4.77 / 3.45

Table 4.7: Statistics about the subsets of the COGS dataset on 33% selected instances based on Inv PPL, CHIA, and BLEU measures. We report average input/output length and word rarity. Statistics are averaged over 3 runs.

splits. Conversely, both *ambiguous* and *easy-to-learn* samples tend to be shorter in length compared to the *randomly selected* samples. Analyzing word rarities, we observe that the *hard-to-learn* subsets have lower input rarity but higher output rarity compared to the *random* subset. On the other hand, the *easy-to-learn* and *ambiguous* samples show higher input rarity than the *random* subset. Notably, the word rarity in *ambiguous* samples surpasses even that of the *hard-to-learn* samples. These statistics provide valuable insights into the subsets. However, determining whether dataset cartography is solely driven by factors such as length and rarity or represents a more complex distribution of samples remains a topic for future investigation.

The statistics of the COGS subsets, as presented in Table 4.7, show similar patterns to the CFQ subsets discussed in Table 4.6. Specifically, we observe that

the *hard-to-learn* subsets tend to have longer samples compared to the *ambiguous* subsets, while the *ambiguous* subsets are longer than the *easy-to-learn* subsets. However, unlike the CFQ subsets, the COGS subsets display an interesting characteristic: as the subsets become harder, there is an increase in the presence of rare words both within and outside the dataset vocabulary. This phenomenon can be attributed to the larger vocabulary size and smaller dataset size of the COGS dataset, as outlined in Table 4.1. Consequently, the variability in word usage plays a more prominent role in determining the hardness of the data instances in COGS. Therefore, by employing dataset cartography, we are able to select subsets that exhibit different underlying factors, ultimately leading to dataset-specific improvements in performance.

$$\text{Rarity}(i) = -\frac{1}{T}\sum_{t=1}^{T}\log f\left(y_{it}^{*}\right) \tag{4.7}$$

4.2.8 Detailed Error Analysis

To gain further insights into the performance of our model, we conduct a comprehensive manual error analysis on both the CFQ and COGS datasets. Our objective was to identify the specific test samples where the model exhibits improved performance after training with a selected subset and to determine the general properties of these samples.

On the CFQ dataset. Our analysis reveals that the *hard-to-learn (Inv PPL)* model outperforms the 100% trained model, particularly on sentences that are shorter than average or of average length. This observation highlights the effectiveness of dataset cartography in enhancing compositional generalization, without relying on spurious correlations. However, it is important to note that the *hard-to-learn (Inv PPL)* model does not demonstrate the same level of improvement in generalizing to longer sentences. Figure 4.6 provides further insights into the performance of the models. We observe a slight increase in errors for the shortest samples. This can be attributed to the fact that the *hard-to-learn (Inv PPL)* model generates longer outputs than the target for a subset of these samples ((1)). This behaviour can be explained by

the length bias present in the *hard-to-learn* subsets, as the models tend to generate longer outputs when encountering instances longer than the average length.

(1) Was a film director M0 →

GOLD: `SELECT count ( * ) WHERE { M0 a film.director }`

OUT: `SELECT count ( * ) WHERE { M0 a film.director . M0 film.director.film M2 }`

Errors observed in both models exhibit a systematic nature. For instance, there are samples where the models fail to correctly order the triple sequences, resulting in incorrect output (see Example (2)). Another common error type involves swapping the 1st argument in a triple with the 3rd argument ((3), formatted for spacing). It is worth noting that these error patterns are not specific to either the *hard-to-learn* or the 100% trained models.

(2) Was M0 ' s prequel a film →

GOLD: `SELECT count ( * ) WHERE { ?x0 a film.film . ?x0 film.film.sequel M0 }`

OUT: `SELECT count ( * ) WHERE { ?x0 film.film.sequel M0 . ?x0 a film.film }`

(3) Was a character M1 ' s director →

GOLD: `SELECT count(*) WHERE { ?x0 a fictional_universe. fictional_character . ?x0 film.director.film M1 }`

OUT: `SELECT count(*) WHERE { ?x0 a fictional_universe. fictional_character . M1 film.director.film ?x0 }`

On the COGS dataset. In the COGS dataset, each instance belongs to one of the 21 generalization categories, such as *Passive* → *Active*, where the verb structure is

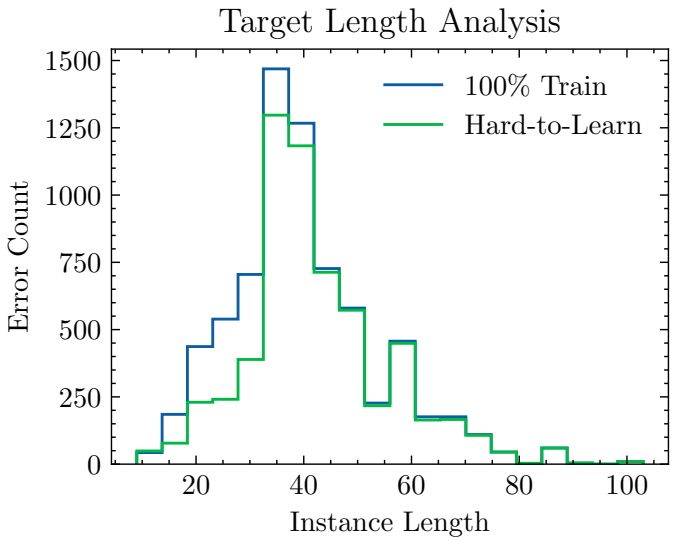

Figure 4.6: Target length histogram for test errors on CFQ.

transformed from a passive form to an active form (e.g. "The book was **squeezed**". → "The girl **squeezed** the strawberry."). This categorization allows us to explore the changes in accuracies across different categories when comparing models trained on 33% *hard-to-learn* subsets and the 100% training dataset (refer to Table 4.8).

For *Inv PPL*, we see performance increase nearly in all of the categories compared to full training except 7 categories. In 3 of these categories, all 4 models have 0% accuracy. And from the remaining 4 categories, in only one category the performance discrepancy is remarkable (*Subject → Object (common noun)*). While dataset cartography significantly contributes towards lexical generalization, the contribution towards structural generalization remains limited.

Among models trained with subsets, *Inv PPL* outperforms *CHIA* on almost all categories, and while overall *Inv PPL* performs better, the ranking between *Inv PPL* and *BLEU* is more volatile. Results in Table 4.8 indicate that the Inv PPL measure can distinguish informative examples better compared to the CHIA measure. Moreover, different confidence measures have different characteristics, therefore they can give more importance to measure-specific instances.

Category	**hard-to-learn (33% train)**			**100% training**
	Inv PPL	**CHIA**	**BLEU**	
Subject → Object (common noun)	0.45	0.57	0.85	0.61
Subject → Object (proper noun)	0.08	0.05	0.16	0.08
Object → Subject (common noun)	0.95	0.95	0.97	0.90
Object → Subject (proper noun)	0.54	0.31	0.40	0.41
Primitive noun → Subject (common noun)	0.93	0.58	0.92	0.63
Primitive noun → Subject (proper noun)	0.90	0.73	0.81	0.44
Primitive noun → Object (common noun)	0.47	0.19	0.14	0.15
Primitive noun → Object (proper noun)	0.19	0.27	0.15	0.14
Primitive verb → Infinitival argument	0.48	0.37	0.07	0.26
Object-modifying PP → Subject-modifying PP	0.00	0.00	0.00	0.00
Depth generalization: Sentential complements	0.00	0.00	0.00	0.00
Depth generalization: PP modifiers	0.09	0.07	0.08	0.07
Active → Passive	0.98	0.94	0.89	0.99
Passive → Active	0.72	0.57	0.74	0.40
Object-omitted transitive → Transitive	0.89	0.65	0.81	0.87
Unaccusative → Transitive	0.46	0.45	0.49	0.41
Double object dative → PP dative	0.55	0.48	0.71	0.54
PP dative → Double object dative	0.65	0.42	0.43	0.17
Agent NP → Unaccusative Subject	0.45	0.48	0.39	0.29
Theme NP → Object-omitted transitive Subject	0.74	0.74	0.85	0.80
Theme NP → Unergative subject	0.71	0.71	0.77	0.78

Table 4.8: COGS accuracy by generalization categories. Subsets have 33% of the original dataset size. Each result is averaged over 3 runs.

Chapter 5

DIVERSE DATASET CARTOGRAPHY

We propose Diverse CARTography (DiCart) that fuses dataset cartography for generative tasks and DPP for a diversity- and complexity-aware subset selection in LLMs. First, we reveal the limitations of directly applying the findings in **the first setup** to the LLM domain. Then, we explore the underlying reasons and remedy the poor performance by innovatively fusing dataset cartography techniques for generative tasks with DPP to curate.

Our approach starts with the standard dataset cartography process, where the training dynamics of an initial LLM are recorded and used to fine-tune a subsequent model. To enhance the effectiveness of this process without incurring additional computational overhead, DiCart emphasizes the importance of diversity in the selected subsets. The selection phase introduces a similarity matrix to create a kernel for DPP, further refined by scaling with a cartography matrix that highlights the importance of particular type of example pairs. This sophisticated scaling mechanism ensures a balanced trade-off between complexity and diversity, achieved through a smoothing operation that adjusts data examples' confidence or variability scores. Through rigorous comparison with several baselines, including random subsets and full training sets, DiCart demonstrates its superior performance and robustness.

5.1 Approach

Dataset cartography is an expensive subset selection technique due to training an initial model with the full training set and a final model with the selected subset. To the best of our knowledge, the initial model is only used for its training dynamics without improving any other aspect of subset selection. Hence, we explore enhancing dataset cartography without overhead, focusing on the diversity of selected subsets.

As Chapter 4 already laid down a systematic comparison between different dataset cartography metrics, we use the best metric from the prior experiments, namely the **Inv PPL** metric throughout this chapter. Hence, we prefer to omit this detail in the following sections for brevity.

5.1.1 Pre-processing

DiCart follows the dataset cartography procedure in Chapter 4.1 until the feature extraction step: the initial LLM's training dynamics are recorded on the full training set during fine-tuning. Similarly, we specify **min epoch** and **convergence epoch** to consider the training dynamics of epochs from **min epoch** until **convergence epoch**. We set **min epoch** to 1 and **convergence epoch** to 5 throughout our experiments in setup 2 without any hyperparameter tuning.

After the fine-tuning, we use the fine-tuned model to extract features for each training instance. We calculate the feature vector of each example as the weighted mean of the hidden representations from the LLM's last layer [Muennighoff, 2022]. The feature vector $\mathbf{v} = \sum_{j=1}^{S} w_j \mathbf{h}_j$ where $w_j = \frac{j}{\sum_{i=1}^{S} i}$ is the weight of the hidden representations from the LLM's last layer $\mathbf{h}_j$ at token j. The weighted mean is harnessed to improve the feature learning hindrances of LLMs due to causal attention. Finally, the vectors are normalized and stored in a feature matrix $\mathbf{V}$.

5.1.2 Subset Selection

Firstly, a cosine similarity matrix is calculated as $\mathbf{S} = \mathbf{V}^T\mathbf{V}$ to create a kernel for DPP. DiCart uses the training dynamics to scale the similarity matrix to increase the probability of subsets including certain data example pairs. For example, by making hard-to-learn example pairs more probable in a subset, we effectively increase the amount of hard-to-learn examples in the subset compared to simply using the similarity matrix. However, as we only scale the similarity matrix, the diversity of selected subsets is still inherent.

To scale the similarity matrix $\mathbf{S}$, we create a cartography matrix $\mathbf{C}$ depending on the subset category (e.g. hard-to-learn). For instance, if we would like to include

more hard-to-learn examples in our selected subset, we store the confidence of each example in the vector $\mathbf{u}$ as hard-to-learn examples exhibit low confidence (see Section 3.2). Then, $\mathbf{C} = \mathbf{u}^T\mathbf{u}$ will have smaller values for harder pairs compared to others. Smaller values in $\mathbf{C}$ will result in smaller values in the final DPP kernel, meaning that harder examples are *less similar*. This will increase the probability of harder pairs being together in the selected subset to improve the diversity.

However, before calculating $\mathbf{C}$, we apply a smoothing operation to $\mathbf{u}$ to control the trade-off between the complexity and diversity of the selected subset. During our experiments, we observed that LLMs can predict large proportions of training sets with total confidence and/or no ambiguity. Therefore, the $\mathbf{C} \odot \mathbf{S}$ operation without smoothing $\mathbf{u}$ renders a big portion of easy-to-learn and non-ambiguous examples impossible to select. Additionally, different setups might require different trade-offs between complexity and diversity, thus the smoothing allows a more flexible framework.

Naturally, the confidence and variability of a data example is within the range $[0, 1]$. As we want to make fringe examples selectable and to control the effect of cartography on diversity, we introduce a smoothing constant α that linearly squashes the confidence or the variability of examples to the range $[\alpha, 1 - \alpha]$. The formula of the transformation is denoted:

$$c = \alpha + (1 - 2\alpha) \times c \tag{5.1}$$

where the range of c (confidence or variability) $[0, 1]$ is linearly pushed to $[\alpha, 1-\alpha]$. We apply this operation to every element of $\mathbf{u}$ and then calculate $\mathbf{C} = \mathbf{u}^T\mathbf{u}$. We set α to 0.1 as default.

Finally, we calculate the kernel matrix as $\mathbf{L} = \mathbf{C} \odot ((\mathbf{S}+1)/2)$. The final $(\cdot+1)/2$ operation converts the similarity matrix to a positive semi-definite matrix, ensuring that $\mathbf{L}$ is positive semi-definite. Then, we sample examples via greedy MAP inference [Chen et al., 2018] until the desired sample size is reached.

5.2 Experimental Setup

While we apply DiCart to select diverse easy-to-learn and ambiguous examples as well, we observe that diverse hard-to-learn subsets perform the best among them. Thus, in this section, DiCart and dataset cartography focus on including hard-to-learn subsets rather than all subset categories.

As mentioned earlier, we aim to improve the compositional generalization of LLMs more efficiently. Therefore, we examine the transfer of training dynamics and features of training examples within our framework.

5.2.1 Baselines

We consider several baselines for our setup. The natural baselines are random subsets of the same size and the full training set. Additionally, we compare DiCart with DPP and the hard-to-learn subset of dataset cartography as the hard-to-learn subset performed the best in our preliminary results.

5.2.2 Datasets

We compare DiCart with our baselines on four datasets, namely ATIS [Hemphill et al., 1990, Dahl et al., 1994], GeoQuery [Zelle and Mooney, 1996], Overnight [Wang et al., 2015], and lastly a simplified version [Meron, 2022] of SMCalFlow-CS (SMCS) dataset [Yin et al., 2021, Andreas et al., 2020].

For ATIS and Overnight, we consider the template splits [Finegan-Dollak et al., 2018] created by [Gupta et al., 2022]. For the GeoQuery dataset, we experiment with three template splits, three TMCD splits, and a length split from [Qiu et al., 2022] – where the results are averaged for brevity. We use the 32-C split for SMCalFlow-CS following [İnce et al., 2023] and [Gupta et al., 2023]. We use **exact match** as our evaluation metric.

5.2.3 Models

We experiment with different model groups to ensure that our results hold across different LLMs. While we mainly report results with Llama 2 7B [Touvron et al.,

Table 5.1: Examples from datasets used in this chapter

Dataset	Source	Target
ATIS	*list flights from la guardia to burbank*	`( lambda $0 e ( and ( flight $0 ) ( from $0 lga : ap ) ( to $0 burbank : ci ) ) )`
GeoQuery	*what is the length of the river that runs through the most number of states*	`answer ( len ( most ( river , traverse_2 , state ) ) )`
Overnight	*employee that has the smallest start date*	`(call listValue (call getProperty ((lambda s (call superlative (var s) (string min) (call ensureNumericProperty (string employment_start_date)))) (call domain (string employee))) (string employee)))`
SMCS	*Schedule a meeting with Lori and Tony today after 1 pm*	`CreateEvent( AND( with_attendee( Lori ) , with_attendee( Tony ) , starts_at( OnDateAfterTime( date= Today( ) , time= NumberPM( 1 ) ) ) ) )`

2023] and Gemma 7B [Team et al., 2024] models, we also use training dynamics from TinyLlama 1.1B [Zhang et al., 2024] and Gemma 2B [Team et al., 2024] respectively. We access the models through HuggingFace and fine-tune models with 16-bit precision (`bfloat16` or `float16` depending on the model) LoRA [Hu et al., 2022] through their PEFT library [Mangrulkar et al., 2022].

5.3 Results

Table 5.2 demonstrates that DiCart is the best-performing method in selecting 50% subsets across different datasets and model families. DiCart especially gains significant performance over baselines in Overnight and SMCS datasets. Even if dataset cartography outperforms DiCart in the ATIS dataset, it fails to be robust across

datasets as it performs worse than the random subset in the Overnight dataset. The synergy between dataset cartography and DPP improves the stability of DiCart, preventing the model from failing terribly. Moreover, DPP is the strongest baseline but still cannot pass DiCart in any setting, showing the non-negligible contribution of dataset cartography in DiCart.

Perhaps more notably, DiCart surpasses the performance of using the full training data in every setting. While DPP also outperforms the 100% setting, dataset cartography cannot outperform the 100% setting in half of the settings.

In the Table 5.3, we see results of 33% subset selection. Overall, DiCart outperforms 100% training performance in half of the settings and is the best subset selection method with DPP. While dataset cartography is the leading method in two settings, it critically fails in the Overnight dataset where it underperforms compared to the random selection. A similar case for DPP appears in the SMCS - Llama 2 7B setting (Subtable 5.3), where DPP severely underperforms compared to other baselines even if it outperforms the random selection. Therefore, DPP might compete with DiCart in multiple settings, it is not as stable as DiCart. This instability raises concerns about the applicability of DPP in other settings and highlights the regularization effect of DPP and dataset cartography over each other.

In addition to the results in 5.2 and 5.3, we report results separately for different training dynamics (and feature extraction) models. This way, we can investigate if incorporating smaller models to collect training dynamics or features hurts the generalization performance. Furthermore, these experiments are crucial for the efficiency concerns of DiCart – and dataset cartography as these methods require training two models sequentially.

We first examine the 50% subset selection setting. In Table 5.4, we do not observe a significant difference between using a smaller model than the model-to-be-trained with DiCart. In the Gemma family of models, collecting dynamics using Gemma 2B results in better performance in SMCS and ATIS while underperforming in Overnight and GeoQuery datasets compared to using Gemma 7B. In contrast, utilizing the dynamics of TinyLlama while training Llama 2 7B results in poorer performance in comparison with using Llama 2 7B for training dynamics in three

Table 5.2: Accuracy results for 50% subset selection. 100% denotes training the model with the full training set while other methods only use 50% of the training data. The best and second-best performing subsets are bolded and underlined, respectively. The results are averaged over experiments where same- and smaller-scale models collect training dynamics and features. The scores are averaged over 10 random seeds and std. dev. is shown as a subscript. The model names are shortened from Gemma 7B and Llama 2 7B for brevity.

Method	Score
DiCart	**41.88**$_{9.22}$
DPP	<u>38.21</u>$_{11.80}$
Cart.	32.24$_{8.15}$
Random	37.56$_{9.90}$
100%	37.55$_{15.43}$

(a) Overnight - Gemma

Method	Score
DiCart	**31.27**$_{7.31}$
DPP	<u>30.39</u>$_{5.62}$
Cart.	25.86$_{7.44}$
Random	27.34$_{9.42}$
100%	28.42$_{9.46}$

(b) Overnight - Llama

Method	Score
DiCart	**52.05**$_{9.64}$
DPP	<u>47.73</u>$_{8.87}$
Cart.	47.05$_{10.71}$
Random	37.58$_{9.71}$
100%	45.79$_{5.69}$

(c) SMCS - Gemma

Method	Score
DiCart	**35.29**$_{8.33}$
DPP	<u>34.52</u>$_{7.25}$
Cart.	29.26$_{7.37}$
Random	22.50$_{8.62}$
100%	33.46$_{4.90}$

(d) SMCS - Llama

Method	Score
DiCart	<u>66.05</u>$_{2.27}$
DPP	65.49$_{3.72}$
Cart.	**66.67**$_{3.44}$
Random	61.21$_{3.32}$
100%	63.44$_{6.09}$

(e) ATIS - Gemma

Method	Score
DiCart	<u>66.09</u>$_{1.92}$
DPP	65.93$_{2.82}$
Cart.	**66.23**$_{2.29}$
Random	62.62$_{1.55}$
100%	65.82$_{0.99}$

(f) ATIS - Llama

Method	Score
DiCart	**67.23**$_{6.64}$
DPP	<u>66.79</u>$_{7.37}$
Cart.	65.58$_{8.24}$
Random	57.73$_{10.32}$
100%	65.69$_{7.01}$

(g) GeoQuery - Gemma

Method	Score
DiCart	**67.48**$_{3.76}$
DPP	<u>66.46</u>$_{4.40}$
Cart.	66.14$_{3.98}$
Random	59.46$_{4.18}$
100%	65.49$_{4.38}$

(h) GeoQuery - Llama

out of four settings.

We observe a similar story in the 33% setting as in the 50% subset selection setting (see Table 5.5). In the Gemma family, harnessing Gemma 2B and Gemma 7B training dynamics are tied as Gemma 2B performs better in Overnight and SMCS while underperforming in other datasets. Using the training dynamics of Llama 2

Table 5.3: Accuracy results for 33% subset selection. 100% denotes training the model with the full training set while other methods only use 33% of the training data. The best and second-best performing subsets are bolded and underlined, respectively. The results are averaged over experiments where same- and smaller-scale models collect training dynamics and features. The scores are averaged over 10 random seeds and std. dev. is shown as a subscript. The model names are shortened from Gemma 7B and Llama 2 7B for brevity.

Method	Score
DiCart	$\mathbf{42.43}_{9.33}$
DPP	$\underline{38.84}_{8.46}$
Cart.	$26.60_{7.49}$
Random	$37.87_{8.20}$
100%	$37.55_{15.43}$

(a) Overnight - Gemma

Method	Score
DiCart	$\mathbf{31.81}_{5.70}$
DPP	$\underline{29.21}_{5.87}$
Cart.	$21.53_{6.70}$
Random	$23.20_{5.25}$
100%	$28.42_{9.46}$

(b) Overnight - Llama

Method	Score
DiCart	$48.78_{9.00}$
DPP	$\mathbf{51.01}_{9.66}$
Cart.	$\underline{50.51}_{8.34}$
Random	$31.67_{8.81}$
100%	$45.79_{5.69}$

(c) SMCS - Gemma

Method	Score
DiCart	$\underline{33.81}_{8.09}$
DPP	$28.96_{7.41}$
Cart.	$\mathbf{34.31}_{7.48}$
Random	$15.05_{8.89}$
100%	$33.46_{4.90}$

(d) SMCS - Llama

Method	Score
DiCart	$62.50_{8.39}$
DPP	$\underline{63.62}_{2.46}$
Cart.	$\mathbf{64.26}_{3.51}$
Random	$59.04_{5.18}$
100%	$63.44_{6.09}$

(e) ATIS - Gemma

Method	Score
DiCart	$\mathbf{63.10}_{1.97}$
DPP	$\underline{62.51}_{3.98}$
Cart.	$62.17_{3.77}$
Random	$60.13_{2.72}$
100%	$65.82_{0.99}$

(f) ATIS - Llama

Method	Score
DiCart	$63.55_{7.55}$
DPP	$\mathbf{65.28}_{6.01}$
Cart.	$\underline{64.59}_{8.26}$
Random	$53.63_{7.86}$
100%	$65.69_{7.01}$

(g) GeoQuery - Gemma

Method	Score
DiCart	$\underline{64.77}_{3.21}$
DPP	$\mathbf{64.99}_{3.26}$
Cart.	$63.56_{4.16}$
Random	$52.95_{4.50}$
100%	$65.49_{4.38}$

(h) GeoQuery - Llama

7B fails to outperform using the dynamics of TinyLlama in two out of four settings.

Thus, we conclude that using larger models for training dynamics results in a limited contribution considering the cost of fine-tuning a larger model. Moreover, these results emphasize that DiCart is robust when utilizing smaller models, simultaneously highlighting the robustness and efficiency of DiCart.

Table 5.4: Accuracy results for 50% subset selection. 100% denotes using the full training set while others use 50% of the set. TD LM denotes the LLM used for training dynamics and feature extraction. The scores are averaged over 10 random seeds and std. dev. is shown as a subscript.

TD LM	Gemma 2B	Gemma 7B
DiCart	$40.97_{10.15}$	$42.80_{8.64}$
DPP	$39.45_{11.93}$	$36.97_{12.18}$
Cart.	$34.51_{10.04}$	$29.96_{5.28}$
Random	$37.56_{9.90}$	
100%	$37.55_{15.43}$	

(a) Overnight - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$31.67_{6.26}$	$30.86_{8.55}$
DPP	$30.53_{6.61}$	$30.25_{4.80}$
Cart.	$27.59_{3.92}$	$24.14_{9.74}$
Random	$27.34_{9.42}$	
100%	$28.42_{9.46}$	

(b) Overnight - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$56.28_{6.14}$	$47.82_{10.89}$
DPP	$50.69_{8.91}$	$44.77_{9.33}$
Cart.	$44.55_{11.04}$	$49.56_{10.30}$
Random	$37.58_{9.71}$	
100%	$45.79_{5.76}$	

(c) SMCS - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$38.22_{7.92}$	$32.36_{8.03}$
Cart.	$29.49_{8.82}$	$29.03_{6.08}$
DPP	$36.98_{6.40}$	$32.05_{7.52}$
Random	$22.50_{8.62}$	
100%	$33.46_{4.90}$	

(d) SMCS - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$66.33_{2.70}$	$65.77_{1.85}$
DPP	$65.91_{3.82}$	$65.06_{3.77}$
Cart.	$66.46_{3.86}$	$66.88_{3.17}$
Random	$61.21_{3.32}$	
100%	$63.44_{6.09}$	

(e) ATIS - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$65.54_{1.86}$	$66.65_{1.89}$
DPP	$65.56_{3.20}$	$66.30_{2.49}$
Cart.	$65.33_{2.73}$	$67.14_{1.32}$
Random	$62.62_{1.55}$	
100%	$65.82_{0.99}$	

(f) ATIS - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$66.19_{7.76}$	$68.27_{5.03}$
DPP	$66.27_{8.67}$	$67.30_{4.02}$
Cart.	$67.13_{4.44}$	$64.03_{10.47}$
Random	$57.73_{10.32}$	
100%	$65.69_{7.01}$	

(g) GeoQuery - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$67.59_{3.96}$	$67.37_{3.55}$
DPP	$67.22_{3.68}$	$65.70_{4.60}$
Cart.	$65.73_{3.53}$	$66.56_{4.33}$
Random	$59.46_{4.18}$	
100%	$65.49_{4.38}$	

(h) GeoQuery - Llama 2 7B

Table 5.5: Accuracy results for 33% subset selection. 100% denotes using the full training set while others use 33% of the set. TD LM denotes the LLM used for training dynamics and feature extraction. The scores are averaged over 10 random seeds and std. dev. is shown as a subscript.

TD LM	Gemma 2B	Gemma 7B
DiCart	$44.33_{9.45}$	$40.53_{9.30}$
DPP	$39.38_{11.23}$	$38.29_{4.93}$
Cart.	$25.83_{5.64}$	$27.38_{9.23}$
Random	$37.87_{8.20}$	
100%	$37.55_{15.43}$	

(a) Overnight - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$31.02_{5.81}$	$32.61_{5.77}$
DPP	$28.96_{4.70}$	$29.45_{7.11}$
Cart.	$22.89_{6.96}$	$20.18_{6.51}$
Random	$23.20_{5.25}$	
100%	$28.42_{9.46}$	

(b) Overnight - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$48.88_{9.92}$	$48.69_{8.52}$
DPP	$51.68_{10.96}$	$50.35_{8.72}$
Cart.	$50.32_{9.66}$	$50.71_{7.31}$
Random	$31.67_{8.81}$	
100%	$45.79_{5.76}$	

(c) SMCS - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$36.31_{5.62}$	$31.31_{9.63}$
DPP	$30.92_{5.72}$	$26.99_{8.64}$
Cart.	$32.82_{5.38}$	$35.79_{9.19}$
Random	$15.05_{8.89}$	
100%	$33.46_{4.90}$	

(d) SMCS - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$60.74_{11.65}$	$64.25_{2.46}$
DPP	$64.09_{2.58}$	$63.14_{2.38}$
Cart.	$63.35_{3.76}$	$65.17_{3.18}$
Random	$59.04_{5.18}$	
100%	$63.44_{6.09}$	

(e) ATIS - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$63.35_{1.82}$	$62.84_{2.17}$
DPP	$63.62_{3.72}$	$61.41_{4.12}$
Cart.	$62.57_{1.90}$	$61.76_{5.10}$
Random	$60.13_{2.72}$	
100%	$65.82_{0.99}$	

(f) ATIS - Llama 2 7B

TD LM	Gemma 2B	Gemma 7B
DiCart	$63.27_{7.18}$	$63.83_{7.89}$
DPP	$64.46_{6.70}$	$66.10_{4.59}$
Cart.	$64.67_{7.44}$	$64.51_{8.72}$
Random	$53.63_{7.86}$	
100%	$65.69_{7.01}$	

(g) GeoQuery - Gemma 7B

TD LM	Llama 2 7B	TinyLlama
DiCart	$64.64_{3.64}$	$64.91_{2.62}$
DPP	$65.20_{2.67}$	$64.78_{3.62}$
Cart.	$63.35_{4.61}$	$63.76_{3.60}$
Random	$52.95_{4.50}$	
100%	$65.49_{4.38}$	

(h) GeoQuery - Llama 2 7B

Chapter 6

CONCLUSION

Transformers are great at language modelling and various downstream tasks, but their ability to achieve compositional generalization compared to humans remains debatable. In this study, we addressed this challenge by demonstrating that selecting a subset of the training dataset using dataset cartography and training models on this subset can enhance model accuracy by up to 10%. We showed that our setup can generalize to different model architectures and natural datasets. Moreover, we achieved improved performance by employing a dataset cartography-based curriculum learning without the need for hyperparameter tuning.

Moreover, we propose DiCart, a diverse and complex subset selection method to improve the compositional generalization of LLMs. DiCart comprises dataset cartography and determinantal point processes (DPP), which facilitates a nuanced balance between complex examples that do not fully cover the training set and diverse examples that are easier but help encompass the scope of the task.

We compare DiCart with full training sets, random subsets, dataset cartography, and DPP on two different model families and four datasets. DiCart outperforms full training sets in every setting with only 50% of the training set and performs on par while using only 33% of the training data.

DiCart outperforms baselines in almost every setting when using 50% of the training data and is one of the best methods under the 33% training data setting. The significance of DiCart is reinforced by its robustness across different settings where other baselines might occasionally perform worse than the random selection.

In this thesis, we make the following contributions to the literature: (1) we propose dataset cartography as a curriculum learning metric, (2) we expand dataset cartography from classification to generative tasks and provide ample analysis of different ways of bridging the distance dataset cartography and generative modelling,

(3) we showcase that harnessing the power of training dynamics for compositional generalization significantly boosts the performance, (4) we introduce DiCart, a new subset selection method based on dataset cartography that also promotes diversity in the subset without any overhead, (5) we demonstrate that DiCart outperforms baselines and even the full training set.

While DiCart highlights an important step in the subset selection domain, it can be improved in several directions. Firstly, DiCart can be integrated within the initial training where the subset filtering occurs after a few epochs before the training finishes. This change has the potential to improve DiCart's performance and carbon footprint. Secondly, reconfiguring smoothing factor α at different subset sizes benefits DiCart to strike the nuanced balance between complexity and diversity. Utilizing a meta-model to eliminate the hyperparameter α can decrease the need for experimentation to achieve the best performance. Thirdly, we can utilize the scores of training examples for sampling where certain examples are sampled more and others are sampled less during the training rather than performing a hard subset selection à la our thesis.

Looking ahead, we anticipate that this research direction promises insights into the necessary syntax, semantics, and structure for informative data instances, informing the development of novel data augmentation strategies and advancing our understanding of deep models' generalization capabilities.

BIBLIOGRAPHY

[Abbas et al., 2023] Abbas, A. K. M., Tirumala, K., Simig, D., Ganguli, S., and Morcos, A. S. (2023). Semdedup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

[Akyurek and Andreas, 2021] Akyurek, E. and Andreas, J. (2021). Lexicon learning for few shot sequence modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4934–4946, Online. Association for Computational Linguistics.

[Albalak et al., 2024] Albalak, A., Elazar, Y., Xie, S. M., Longpre, S., Lambert, N., Wang, X., Muennighoff, N., Hou, B., Pan, L., Jeong, H., Raffel, C., Chang, S., Hashimoto, T., and Wang, W. Y. (2024). A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*. https://arxiv.org/abs/2402.16827.

[An et al., 2023] An, S., Lin, Z., Fu, Q., Chen, B., Zheng, N., Lou, J.-G., and Zhang, D. (2023). How do in-context examples affect compositional generalization? In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11027–11052, Toronto, Canada. Association for Computational Linguistics.

[Andreas, 2020] Andreas, J. (2020). Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

[Andreas et al., 2020] Andreas, J., Bufe, J., Burkett, D., Chen, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., Fang, H., Guo, A., Hall, D., Hayes, K., Hill, K., Ho, D., Iwaszuk, W., Jha, S., Klein, D., Krishnamurthy, J., Lanman, T., Liang, P., Lin, C. H., Lintsbakh, I., McGovern, A., Nisnevich, A., Pauls, A., Petters, D., Read, B., Roth, D., Roy, S., Rusak, J., Short, B., Slomin, D., Snyder, B., Striplin, S., Su, Y., Tellman, Z., Thomson, S., Vorobev, A., Witoszko, I., Wolfe, J., Wray, A., Zhang, Y., and Zotov, A. (2020). Task-Oriented Dialogue as Dataflow Synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

[Andreas et al., 2016] Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016). Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

[Azeemi et al., 2023] Azeemi, A., Qazi, I., and Raza, A. (2023). Data pruning for efficient model pruning in neural machine translation. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 236–246, Singapore. Association for Computational Linguistics.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Bhatnagar et al., 2022] Bhatnagar, R., Ganesh, A., and Kann, K. (2022). CHIA: choosing instances to annotate for machine translation. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 7299–7315. Association for Computational Linguistics.

[Bogin et al., 2022] Bogin, B., Gupta, S., and Berant, J. (2022). Unobserved local structures make compositional generalization hard. In Goldberg, Y., Kozareva,

Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2731–2747, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (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.

[Bugliarello and Elliott, 2021] Bugliarello, E. and Elliott, D. (2021). The role of syntactic planning in compositional image captioning. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 593–607, Online. Association for Computational Linguistics.

[Chen et al., 2018] Chen, L., Zhang, G., and Zhou, E. (2018). Fast greedy map inference for determinantal point process to improve recommendation diversity. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Chen et al., 2020] Chen, X., Liang, C., Yu, A. W., Song, D., and Zhou, D. (2020). Compositional generalization via neural-symbolic stack machines. *Advances in Neural Information Processing Systems*, 33:1690–1701.

[Csordás et al., 2021] Csordás, R., Irie, K., and Schmidhuber, J. (2021). The devil is in the detail: Simple tricks improve systematic generalization of transformers. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 619–634. Association for Computational Linguistics.

[Dahl et al., 1994] Dahl, D. A., Bates, M., Brown, M. K., Fisher, W. M., Hunicke-Smith, K., Pallett, D. S., Pao, C., Rudnicky, A., and Shriberg, E. (1994). Expanding the scope of the atis task: The atis-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

[de Vries et al., 2019] de Vries, H., Bahdanau, D., Murty, S., Courville, A. C., and Beaudoin, P. (2019). CLOSURE: assessing systematic generalization of CLEVR models. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*.

[Devlin et al., 2019] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

[Drozdov et al., 2022] Drozdov, A., Schärli, N., Akyürek, E., Scales, N., Song, X., Chen, X., Bousquet, O., and Zhou, D. (2022). Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*.

[Dziri et al., 2023] Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D., Sanyal, S., Welleck, S., Ren, X., Ettinger, A., Harchaoui, Z., and Choi, Y. (2023). Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

[Finegan-Dollak et al., 2018] Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., and Radev, D. (2018). Improving text-to-SQL evaluation methodology. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

[Fodor and Pylyshyn, 1988] Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

[Furrer et al., 2020] Furrer, D., van Zee, M., Scales, N., and Schärli, N. (2020). Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

[Geach, 1965] Geach, P. (1965). Logical procedures and the identity of expressions. *Ratio (Misc.)*, 7(2):199–205.

[Gupta et al., 2023] Gupta, S., Gardner, M., and Singh, S. (2023). Coverage-based example selection for in-context learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics.

[Gupta et al., 2022] Gupta, S., Singh, S., and Gardner, M. (2022). Structurally diverse sampling for sample-efficient training and comprehensive evaluation. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4966–4979, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Hacohen and Weinshall, 2019] Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

[Hemphill et al., 1990] Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. (1990). The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

[Hu et al., 2022] Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[Hudson and Manning, 2018] Hudson, D. A. and Manning, C. D. (2018). Compositional attention networks for machine reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Hupkes et al., 2020] Hupkes, D., Dankers, V., Mul, M., and Bruni, E. (2020). Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

[İnce et al., 2023] İnce, O., Zeraati, T., Yagcioglu, S., Yaghoobzadeh, Y., Erdem, E., and Erdem, A. (2023). Harnessing dataset cartography for improved compositional generalization in transformers. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13023–13041, Singapore. Association for Computational Linguistics.

[Johnson et al., 2017] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society.

[Keysers et al., 2020] Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., and Bousquet, O. (2020). Measuring compositional generalization: A comprehensive method on realistic data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[Kim and Linzen, 2020] Kim, N. and Linzen, T. (2020). COGS: A compositional

generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

[Kim et al., 2022] Kim, N., Linzen, T., and Smolensky, P. (2022). Uncontrolled lexical exposure leads to overestimation of compositional generalization in pretrained models.

[Kulesza et al., 2012] Kulesza, A., Taskar, B., et al. (2012). Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

[Kumar and Talukdar, 2021] Kumar, S. and Talukdar, P. (2021). Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4507–4518.

[Lake, 2019] Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9788–9798.

[Lake and Baroni, 2017] Lake, B. M. and Baroni, M. (2017). Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks. In *6th International Conference on Learning Representations, ICLR 2018 Workshop Track, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, volume abs/1711.00350. OpenReview.net.

[Lake and Baroni, 2023] Lake, B. M. and Baroni, M. (2023). Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121.

[Le Bras et al., 2020] Le Bras, R., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M., Sabharwal, A., and Choi, Y. (2020). Adversarial filters of dataset biases. In *International conference on machine learning*, pages 1078–1088. Pmlr.

[Levy et al., 2023] Levy, I., Bogin, B., and Berant, J. (2023). Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422.

[Li et al., 2019] Li, Y., Zhao, L., Wang, J., and Hestness, J. (2019). Compositional generalization for primitive substitutions. 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 4293–4302, Hong Kong, China. Association for Computational Linguistics.

[Liu et al., 2020] Liu, Q., An, S., Lou, J.-G., Chen, B., Lin, Z., Gao, Y., Zhou, B., Zheng, N., and Zhang, D. (2020). Compositional generalization by learning analytical expressions. *Advances in Neural Information Processing Systems*, 33:11416–11427.

[Longpre et al., 2024] Longpre, S., Yauney, G., Reif, E., Lee, K., Roberts, A., Zoph, B., Zhou, D., Wei, J., Robinson, K., Mimno, D., and Ippolito, D. (2024). A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

[Maharana et al., 2023] Maharana, A., Yadav, P., and Bansal, M. (2023). D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *arXiv preprint arXiv:2310.07931*.

[Mangrulkar et al., 2022] Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. (2022). Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

[Marcus, 2001] Marcus, G. F. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. The MIT Press.

[McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

[Meron, 2022] Meron, J. (2022). Simplifying semantic annotations of smcalflow. In *Proceedings of the 18th Joint ACL-ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 81–85.

[Muennighoff, 2022] Muennighoff, N. (2022). Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

[Muennighoff et al., 2024] Muennighoff, N., Rush, A., Barak, B., Le Scao, T., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., and Raffel, C. A. (2024). Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

[Ontanon et al., 2022] Ontanon, S., Ainslie, J., Fisher, Z., and Cvicek, V. (2022). Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

[Oren et al., 2021] Oren, I., Herzig, J., and Berant, J. (2021). Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization. In Moens, M.-F., Huang, X., Specia, L., and Yih, S.

W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10793–10809, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

[Partee et al., 1995] Partee, B. et al. (1995). Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360.

[Patel et al., 2022] Patel, A., Bhattamishra, S., Blunsom, P., and Goyal, N. (2022). Revisiting the compositional generalization abilities of neural sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 424–434, Dublin, Ireland. Association for Computational Linguistics.

[Qiu et al., 2022] Qiu, L., Shaw, P., Pasupat, P., Nowak, P., Linzen, T., Sha, F., and Toutanova, K. (2022). Improving compositional generalization with latent structure and data augmentation. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

[Rubin et al., 2022] Rubin, O., Herzig, J., and Berant, J. (2022). Learning to retrieve prompts for in-context learning. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

[Russin et al., 2020] Russin, J., Jo, J., O'Reilly, R., and Bengio, Y. (2020). Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 313–327, Online. Association for Computational Linguistics.

[Sachdeva et al., 2024] Sachdeva, N., Coleman, B., Kang, W.-C., Ni, J., Hong, L., Chi, E. H., Caverlee, J., McAuley, J., and Cheng, D. Z. (2024). How to train data-efficient llms.

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[Smolensky, 1988] Smolensky, P. (1988). The constituent structure of connectionist mental states: A reply to fodor and pylyshyn. *Southern Journal of Philosophy*, 26(S1):137–161.

[Swayamdipta et al., 2020] Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. (2020). Dataset cartography: Mapping and diagnosing datasets with training dynamics. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

[Team et al., 2024] Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Sessa, P. G., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanova, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C. L., Choquette-Choo, C. A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G.-C., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J.-B., Stanway, J., Brennan, J., Chen, J.,

Ferret, J., Chiu, J., Mao-Jones, J., Lee, K., Yu, K., Millican, K., Sjoesund, L. L., Lee, L., Dixon, L., Reid, M., Mikuła, M., Wirth, M., Sharman, M., Chinaev, N., Thain, N., Bachem, O., Chang, O., Wahltinez, O., Bailey, P., Michel, P., Yotov, P., Chaabouni, R., Comanescu, R., Jana, R., Anil, R., McIlroy, R., Liu, R., Mullins, R., Smith, S. L., Borgeaud, S., Girgin, S., Douglas, S., Pandya, S., Shakeri, S., De, S., Klimenko, T., Hennigan, T., Feinberg, V., Stokowiec, W., hui Chen, Y., Ahmed, Z., Gong, Z., Warkentin, T., Peran, L., Giang, M., Farabet, C., Vinyals, O., Dean, J., Kavukcuoglu, K., Hassabis, D., Ghahramani, Z., Eck, D., Barral, J., Pereira, F., Collins, E., Joulin, A., Fiedel, N., Senter, E., Andreev, A., and Kenealy, K. (2024). Gemma: Open models based on gemini research and technology.

[Tirumala et al., 2024] Tirumala, K., Simig, D., Aghajanyan, A., and Morcos, A. (2024). D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36.

[Toneva et al., 2019] Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. (2019). An empirical study of example forgetting during deep neural network learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, Conference Track Proceedings*. OpenReview.net.

[Touvron et al., 2023] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Wang et al., 2019] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

[Wang et al., 2022] Wang, H., Huang, W., Wu, Z., Tong, H., Margenot, A. J., and He, J. (2022). Deep active learning by leveraging training dynamics. *Advances in Neural Information Processing Systems*, 35:25171–25184.

[Wang et al., 2015] Wang, Y., Berant, J., and Liang, P. (2015). Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342.

[Wei et al., 2022] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

[Wei et al., 2013] Wei, K., Liu, Y., Kirchhoff, K., and Bilmes, J. (2013). Using document summarization techniques for speech data subset selection. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 721–726.

[Weißenhorn et al., 2022] Weißenhorn, P., Donatelli, L., and Koller, A. (2022). Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

[Wenzek et al., 2020] Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (2020). CCNet: Extracting high quality monolingual datasets from web crawl data. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of*

the Twelfth Language Resources and Evaluation Conference, pages 4003–4012, Marseille, France. European Language Resources Association.

[Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

[Yagcioglu et al., 2024] Yagcioglu, S., İnce, O. B., Erdem, A., Erdem, E., Elliott, D., and Yuret, D. (2024). Sequential compositional generalization in multimodal models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5591–5611.

[Ye et al., 2023] Ye, J., Wu, Z., Feng, J., Yu, T., and Kong, L. (2023). Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

[Yin et al., 2021] Yin, P., Fang, H., Neubig, G., Pauls, A., Platanios, E. A., Su, Y., Thomson, S., and Andreas, J. (2021). Compositional generalization for neural semantic parsing via span-level supervised attention. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online. Association for Computational Linguistics.

[Zelle and Mooney, 1996] Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

[Zhang et al., 2024] Zhang, P., Zeng, G., Wang, T., and Lu, W. (2024). Tinyllama: An open-source small language model.

[Zhang et al., 2019] Zhang, X., Shapiro, P., Kumar, G., McNamee, P., Carpuat, M., and Duh, K. (2019). Curriculum learning for domain adaptation in neural machine translation. 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 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Dataset	d_{model}	d_{ff}	n_{head}	n_{layer}	batch size	learning rate	warmup	scheduler	n_{param}
CFQ	128	256	16	2	1024	0.9	4000	Noam	685k
COGS	512	512	4	2	128	10^{-4}	-	-	9.3M

Table A.1: Hyperparameters and number of parameters for each task. Feedforward size is denoted as d_{ff}. Only CFQ batch size is changed from [Csordás et al., 2021] (4096 $\rightarrow$ 1024).

Appendix A

DATA MAPS FOR GENERATIVE TASKS

A.1 Reproducibility

We use the experimental setup created in [Csordás et al., 2021] for vanilla Transformers and modify it to calculate and store training dynamics, and implement a curriculum learning framework. After storing training dynamics such as perplexity, CHIA, and BLEU, we choose a subset with criteria. As we choose random subsets as a baseline, we specify the seed during this process. For the Bi-LSTM with attention experiments, we adopt the setup created in [Patel et al., 2022].

As mentioned in the thesis, we used models and hyperparameters from [Csordás et al., 2021] to ease computational constraints and use an initial strong random baseline (see Table A.1). Accuracy is calculated on the sequence level, meaning that all tokens in the output sequence should match all tokens in the gold sequence while preserving the sequence. We use the NLTK BLEU-4 score as the BLEU metric. We specifically use `SmoothingFunction.method4` for smoothing and `auto_reweigh` is set to `True` as some examples are shorter than 4 words. For the CFQ dataset, we applied the same preprocessing as in [Csordás et al., 2021], which is used in [Keysers et al., 2020] as well. For COGS, no dataset preprocessing was used.

A.2 Subset Examples

In the following, we randomly sample examples from 5% hardest-to-learn, most ambiguous, or easiest-to-learn examples. We show examples based on BLEU measure for the CFQ dataset (examples (1), (2), and (3)), and based on Inv PPL measure for the COGS dataset (examples (4), (5), and (6)), for brevity. While these examples are too small for any inference, we see that these examples reflect subset statistics mentioned in Tables 4.6 and 4.7.

CFQ Subset Samples:

(1) *An easy-to-learn sample:*

What did a child of M0 executive produce, edit, write, direct, and produce →

```
SELECT DISTINCT ?x0 WHERE { ?x0 film.film.directed_by ?x1 . s?x0
film.film.edited_by ?x1 . ?x0 film.film.executive_produced_by ?x1
. ?x0 film.film.produced_by| ns:film.film.production_companies
?x1 . ?x0 film.film.written_by ?x1 . ?x1 people.person.parents|
ns:fictional_universe. fictional_character.parents|
ns:organization.organization. parent/ns:organization.
organization_relationship.parent M0 }
```

(2) *An ambiguous sample:*

What did M0 found and M1's female founder found →

```
SELECT DISTINCT ?x0 WHERE { ?x0
organization.organization.founders ?x1 . ?x0
organization.organization.founders M0 . ?x1 organization.
organization_founder. organizations_founded M1 . ?x1
people.person.gender m_02zsn }
```

(3) *A hard-to-learn sample:*

Was M2 a film producer that employed a spouse of M1, employed M0's executive producer, and employed M4 →

```
SELECT count (*) WHERE { ?x0 film.producer.
films_executive_produced M0 . ?x1 people.person.spouse_s/
ns:people.marriage.spouse| ns:fictional_universe.
fictional_character.married_to/ ns:fictional_universe.
marriage_of_fictional_ characters.spouses M1 . FILTER ( ?x1 != M1
) . M2 a film.producer . M2 business.employer.employees/
ns:business.employment_tenure. person ?x0 . M2
business.employer.employees/ ns:business.employment_tenure.
person ?x1 . M2 business.employer.employees/
ns:business.employment_tenure. person M4 }
```

COGS Subset Samples:

(4) *An easy-to-learn sample:*

A cake was drawn by Emma . →

```
cake ( x _ 1 ) AND draw . theme ( x _ 3 , x _ 1 ) AND draw . agent
( x _ 3 , Emma )
```

(5) *An ambiguous sample:*

The cat wished to sleep . →

```
* cat ( x _ 1 ) ; wish . agent ( x _ 2 , x _ 1 ) AND wish . xcomp (
x _ 2 , x _ 4 ) AND sleep . agent ( x _ 4 , x _ 1 )
```

(6) *A hard-to-learn sample:*

James gave a lion a cake in the fridge . →

```
* fridge ( x _ 8 ) ; give . agent ( x _ 1 , James ) AND give .
```

```
recipient ( x _ 1 , x _ 3 ) AND give . theme ( x _ 1 , x _ 5 ) AND
lion ( x _ 3 ) AND cake ( x _ 5 ) AND cake . nmod . in ( x _ 5 ,
x _ 8 )
```

A.3 Remaining Cartography Plots

We present the remaining cartography plots for the CFQ and COGS datasets in this section. Same as previous plots, we only plot randomly sampled 33% of the training set. For the CFQ dataset, the remaining plot is the CHIA plot (Figure A.1). For the COGS dataset, the remaining plots consist of the BLEU plot (Figure A.2) and the CHIA plot (Figure A.3).

Upon examining these plots, we observe distinct characteristics among them. The CHIA plots appear denser, with data examples concentrated in specific regions. In contrast, the BLEU plots exhibit a more widespread distribution, while the Inv PPL plots demonstrate the highest degree of dispersion. These plots offer interesting insights when comparing the performances of CHIA and Inv PPL measures. As instances in Inv PPL plots are better distributed compared to instances in CHIA plots, categories of examples are more distinguishable, resulting in CHIA *hard-to-learn* subsets including *ambiguous* or even *easy-to-learn* instances. Despite their similar underlying mathematical intuition, these plots contribute to a better understanding of the observed differences.

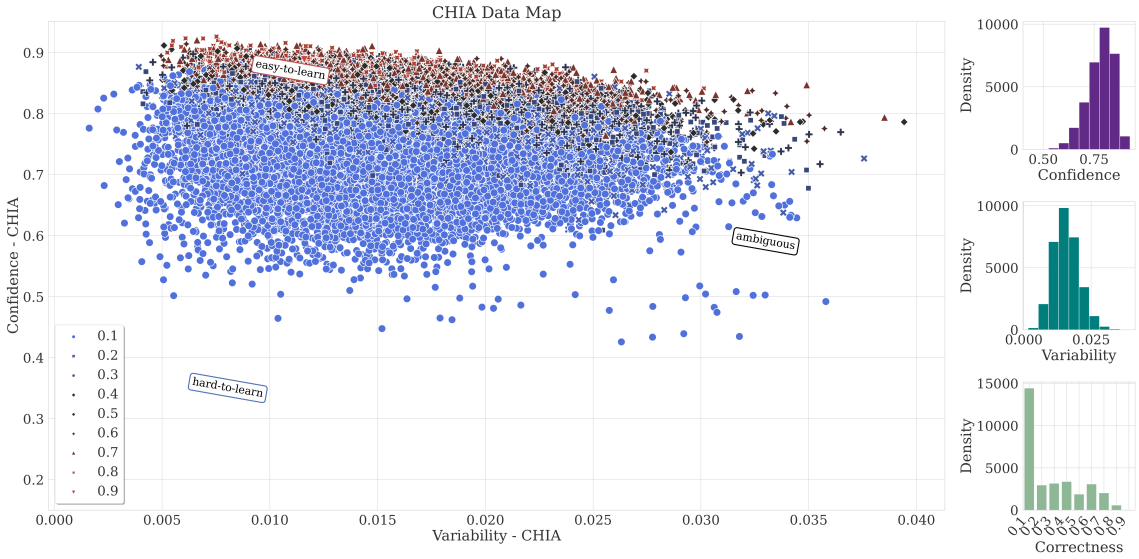

Figure A.1: **Data map of CFQ train set** for the Transformer model based on CHIA measure (converge epoch 20). The x-axis shows the **variability** and the y-axis the **confidence**. The colours and shapes indicate the **correctness**.

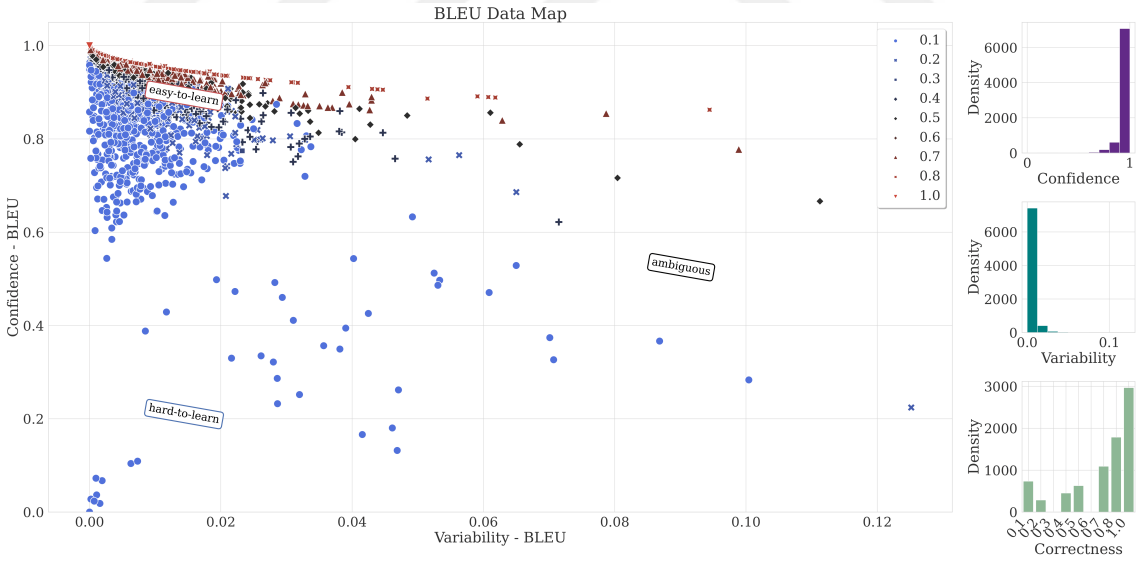

Figure A.2: **Data map of COGS train set** for the Transformer model based on BLEU measure (converge epoch 10). The x-axis shows the **variability** and the y-axis the **confidence**. The colours and shapes indicate the **correctness**.

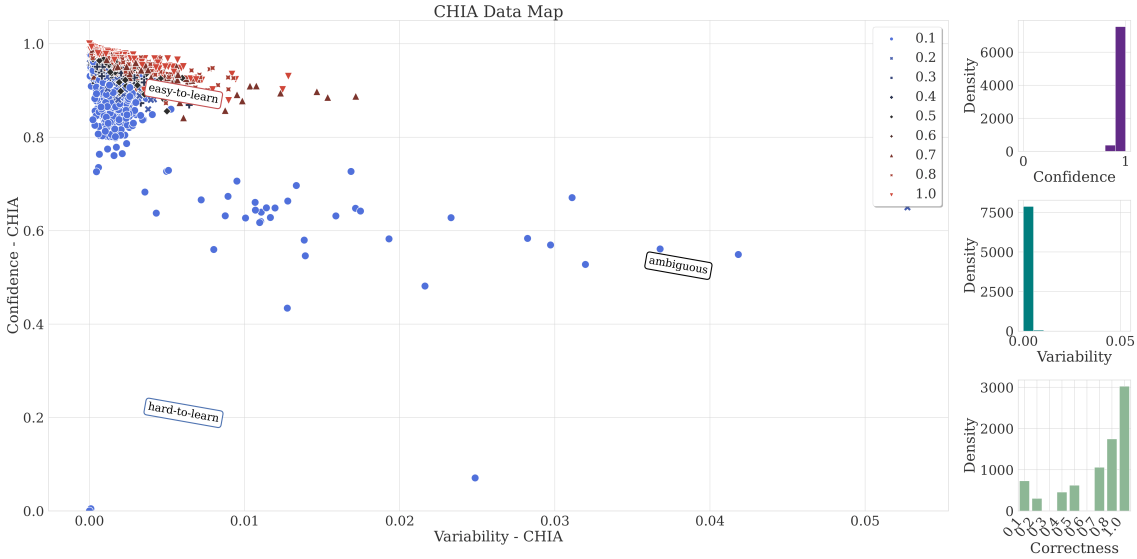

Figure A.3: **Data map of COGS train set** for the Transformer model based on CHIA measure (converge epoch 10). The x-axis shows the **variability** and the y-axis the **confidence**. The colours and shapes indicate the **correctness**.

Appendix B

DIVERSE DATASET CARTOGRAPHY

B.1 Reproducibility

We execute our experiments in a `Python 3.10.14` environment with `PyTorch 2.2.0` deep learning library. We use Huggingface Transformers [Wolf et al., 2020] with Unsloth[1] library for faster fine-tuning. We use `meta-llama/Llama-2-7b-hf` weights for Llama 2 7B, `TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T` for TinyLlama 1.1B, `google/gemma-2b` for Gemma 2B, and `google/gemma-7b` for Gemma 7B models.

We fine-tune these LLMs with parameter-efficient fine-tuning (PEFT) techniques due to computational and temporal concerns. We employ LoRA [Hu et al., 2022] as it is a prevalent and well-researched PEFT method. We add LoRA weights to all weight matrices (query, key, value, out, gate, up, and down projection matrices) to diminish the effect of selected weight matrices on results. Simultaneously, we perform hyperparameter tuning for each model and dataset splits to ensure strong 100% training baselines and more valuable training dynamics. We sample 5% of the training set as the validation set to perform hyperparameter tuning. As our setup has a lot of experiments, we limit the hyperparameter tuning space to the learning rate, LoRA rank (r) parameter, and LoRA α parameter (see Table B.1).

After the hyperparameter tuning, we clone the best configuration into 10 individual experiments where each one has a random seed. Then, we collect training dynamics and extract features for each experiment separately.

While it is not a tuned hyperparameter, we experiment with another method feature, named *primitive coverage*. In this context, primitive means every distinct word in a dataset when the dataset is separated by whitespace. With this feature,

[1] `https://github.com/unslothai/unsloth`

LoRA α	LoRA r	learning rate	batch size	dropout	weight decay
$[8, 16]$	$[16, 32]$	$[1e-4, 3e-4]$	32	0.0	0.0

Table B.1: Hyperparameter space for each task. The best configuration over a single random seed is selected to be followed at the following steps.

we specify if we want to include every primitive from the training set to our chosen subset. We enable *primitive coverage* in our 50% experiments for DiCart & baselines to increase the methods' robustness and disable it for our 33% experiments to allow a freer subset selection.

LLMs can be started to train with the full learning rate after a small number of warmup steps. Nonetheless, we prefer to warm up LLMs for an epoch so that the effect of dataset order on training dynamics reduces. We linearly warm up LLMs for an entire epoch and keep the learning rate constant till the end of the training.

All experiments are performed on a single NVIDIA A40 GPU with CUDA version 11.8. As we want to have a large effective batch size, we use gradient accumulation steps of 4 (batch size of 8) to keep the effective batch size high in Llama 2 7B, Gemma 7B, and Gemma 2B models.

We evaluate our LLMs with the exact match metric, where the prediction and target texts should exactly match. We train our models with the `{source}\t{target}` format where `{source}` and `{target}` are the source and the target of a training example. After training, we prompt the LLMs with `{source}\t`, and perform greedy decoding to collect predictions.

For the UMAP figures (Figure 1.4), we use the `umap-learn` [McInnes et al., 2018] library. We set the `metric=cosine`, `min_dist=0.5`, and `n_neighbors=50` as we want to capture global dependencies and we construct the similarity matrix with cosine similarity before calculating the DPP kernel.

Occasionally, we observed that the greedy MAP sampling process in DPP and DiCart got stuck due to training embeddings being too close to each other in the embedding space for DPP. To alleviate this problem, we take the power of each

element in the similarity matrix incrementally. The power constant starts at 1.0 and is increased by 0.5 after each failed sampling attempt until either the sampling is completed or the constant hits 10.0. If the constant hits 10.0 and the sampling process is still unsuccessful, the accuracy of the run is counted as 0.0.