Inducing Systematicity in Transformers by Attending to Structurally Quantized Embeddings

Yichen Jiang **Xiang Zhou** **Mohit Bansal**
UNC Chapel Hill
{yichenj, xzh, mbansal}@cs.unc.edu

Abstract

Transformers generalize to novel compositions of structures and entities after being trained on a complex dataset, but easily overfit on datasets of insufficient complexity. We observe that when the training set is sufficiently complex, the model encodes structurally equivalent sentences using a systematic attention pattern. Inspired by this observation, we propose *SQ-Transformer* (**S**tructurally **Q**uantized) that explicitly encourages systematicity in the embeddings and attention layers even with low-complexity data. At the embedding level, we introduce Structure-oriented Vector Quantization (SoVQ) to cluster word embeddings into several classes of structurally equivalent entities. At the attention level, we devise the Systematic Attention Layer (SAL) and an alternative, Systematically Regularized Layer (SRL) that operate on the quantized word embeddings so that sentences of the same structure are encoded with invariant or similar attention patterns. Empirically, we show *SQ-Transformer* achieves stronger compositional generalization than the vanilla Transformer on multiple low-complexity semantic parsing and machine translation datasets. In our analysis, we show SoVQ indeed learns a syntactically clustered embedding space, and SAL/SRL induces generalizable attention patterns, altogether leading to improved systematicity.[1]

1 Introduction

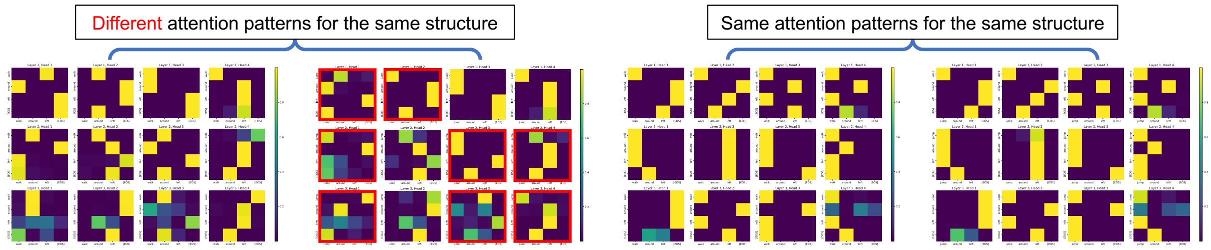
Natural languages demonstrate *compositionality*, which states that the meaning of a complex expression is determined by its syntactic structure and the meanings of its lexical constituents (Chomsky, 1957; Montague, 1970). It leads to humans' algebraic capacity to *systematically* understand a potentially infinite number of novel combinations of known structures and entities. For example, someone who understands "*The cat is asleep*" and "*The dog is awake*" must simultaneously understand "*The dog is asleep*" and "*The cat is awake*".

Early works argued that neural networks are associative devices that cannot capture compositionality (Fodor and Pylyshyn, 1988; Marcus, 1998) and are supported by the empirical results that a Transformer (Vaswani et al., 2017) trained to parse "*walk twice*", "*walk around left*", and '*jump*' fails to parse "*jump twice*" and "*jump around left*" in SCAN ADDJUMP (Lake and Baroni, 2018). Later works presented a more promising picture: for example, Zhou et al. (2023) found that Transformers trained on an augmented, high-complexity dataset with more examples and diverse entities/structures can systematically generalize to novel compositions in SCAN ADDJUMP. Studies on large pretrained models (Furrer et al., 2020; Drozdov et al., 2023) also reveal their ability to systematically generalize. However, data augmentation requires domain-specific knowledge and pretraining on large datasets is also prohibitively expensive. Therefore, how to induce systematicity with low-complexity data has a significant value for improving the model's data efficiency, and remains an open and important research question.

To understand the emergence of systematicity in Transformers, we start by analyzing the attention maps from models trained on SCAN ADDJUMP data of different complexities. First, we demonstrate that a Transformer trained on the original, low-complexity training set (with only 4 primitives) uses different attention weights to encode in-distribution training sentences like "*walk around left*" (Fig. 1a) and an unobserved sentence "*jump around left*" (Fig. 1b). It only achieves 3.7% test accuracy in unobserved sentences. In contrast, the same model trained on large augmented data (with 84 distinct primitives like '*walk*' and '*jump*') uses highly similar attention patterns to encode these two sentences (see Fig. 1c and Fig. 1d). This model

[1] Our code is publicly available at https://github.com/jiangycTarheel/SQ-Transformer.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8360–8383
August 11-16, 2024 ©2024 Association for Computational Linguistics



(a) Attention maps encoding "*walk around left*", trained on original ADDJUMP.

(b) Attention maps encoding "*jump around left*", trained on original ADDJUMP.

(c) Attention maps encoding "*walk around left*", trained on 20x augmented ADDJUMP.

(d) Attention maps encoding "*jump around left*", trained on 20x augmented ADDJUMP.

Figure 1: Attention maps encoding a training example "*walk around left*" and a test example "*jump around left*". The maps in (a) and (b) are from the Transformers trained on the original SCAN ADDJUMP training set; the maps in (c) and (d) are from the model trained on 20x augmented training set (Zhou et al., 2023) with 20 times more primitives like '*walk1*' (mutated from '*walk*') and more examples like "*walk1 around left*". In all attention maps, brighter yellow squares correspond to an attention value close to 1 and darker purple squares correspond to an attention value close to 0. We highlight the attention maps in (b) that differ from (a) in **red boxes**. When trained on 20x augmented training set, the model encodes the two examples with highly similar attention maps across all layers and heads (c and d). We show the attention maps on other examples of the structure "$\$x$ *around left*" in Fig. 5.

achieves 100% test accuracy and encodes the syntactic structure "$\$x$ *around left*" with a unified attention pattern invariant to the choice of $\$x$, as long as $\$x$ has the same syntactic function (e.g., being a verb). Over the entire test set, we observe many such reused attention.[2] Therefore, we test the hypothesis that this ability to systematically reuse learned attention patterns on novel sentences is critical for a Transformer to systematically generalize.

To this end, we propose *SQ-Transformer* with two improvements to the embeddings and attention layers respectively, so as to induce the same systematicity seen above, even with low-complexity training data. The first improvement brings linguistic categorization to the word embeddings: as Johnson (2004) argues *"the claim that natural languages are systematic presupposes a natural non-overlapping linguistic categorization of all the expressions."* For example, for the model to generalize to the unseen "*jump twice*", it first has to learn that '*jump*' belongs to the same category as other primitives like '*walk*' and '*run*'. Motivated by this theory and the finding, we propose **S**tructure-**o**riented **V**ector **Q**uantization (**SoVQ**) to actively cluster all word embeddings into a fixed number of structural equivalence classes, and quantize each word into a code embedding shared among an entire class. We introduce a variational and generalized Brown Clustering (Brown et al., 1992) objective. This unsupervised objective encourages a "**predictive clustering**" such that the class of a token can be predicted by the classes of its context tokens, and hence ensures that words of the same syntactic function are in the same class. After being trained on examples like "*walk*", "*walk around left*", and "*jump*", SoVQ can quantize '*jump*' and '*walk*' into the same class with a code embedding encoding their shared role in a sentence structure.

Our second improvement encourages a unified attention pattern for encoding sentences of a common syntactic structure. The general belief in cognitive science states that *systematicity involves a capacity to represent common structural relations among the equivalently cognizable entities* (Phillips and Wilson, 2016). That is, a systematic mind can always represent a structure even if one or more of its entities is substituted with any equivalently cognizable entity.[3] Since **SoVQ** has quantized each class of equivalently cognizable entities into a code embedding, we then propose the **S**ystematic **A**ttention **L**ayer (**SAL**) that uses these code embeddings as the queries and keys, and the word embeddings as the values (Fig. 2a). When encoding sentences with a common syntactic structure like "$\$x$ *around left*", SAL is **hard-invariant** for any $\$x$ in a structural equivalence class C established by SoVQ. It thus enables the Transformer to systematically represent common structural relations among those quantized classes of equivalently cognizable entities.

To retain the attention's ability to represent non-structural relations that commonly exist in natural

[2] We discuss a quantified analysis in Sec. 5.3.

[3] The notion of "equivalently cognizable entities" generally refers to entities within an equivalence class with respect to certain structural equivalence (e.g., all proper nouns).

languages, we also introduce an alternative to SAL: **S**ystematically **R**egularized **L**ayer (**SRL**). It inherits the architecture of a regular attention layer, but additionally minimizes the L2 distance between the layers' outputs computed from word embeddings and the layers' outputs computed from quantized word embeddings (Fig. 2b). Therefore, unlike SAL, SRL encourages attention's **soft invariance** to structurally equivalent entities: sentences with common structures are processed with **similar** but not necessarily the same attention pattern. Overall, we name this model with SoVQ and SAL/SRL as (**S**tructurally **Q**uantized) ***SQ-Transformer***.

To demonstrate that predictive clustering in embeddings and invariance in attention can lead to systematicity in the model's predictions, we train and evaluate *SQ-Transformer* from scratch on multiple low-complexity semantic parsing and machine translation datasets requiring compositional generalization. In semantic parsing, *SQ-Transformer* improves upon Transformer on SCAN AddJump x2 (Jiang et al., 2022) (40%→99.4%), AroundRight (Loula et al., 2018) (69.5%→99.6%), and COGS (Kim and Linzen, 2020) (82.6%→83.4%). In machine translation, *SQ-Transformer* achieves higher BLEU scores (60.5→62.8) and lower novel compound translation error (29.6%→18.1%) on CoGnition (Li et al., 2021). Importantly, it also shows generalizability to higher-complexity, natural datasets that do not have a significant distribution shift between training and test sets: in WMT En↔De and En↔Fr, *SQ-Transformer* with SRL obtains significantly higher BLEU scores. We further analyze *SQ-Transformer* and present two findings: (1) SoVQ can more effectively cluster word embeddings based on their syntactic functions compared to VQ; (2) SAL and SRL learn attention patterns that can systematically encode unseen compositions of structure and entities. These analyses explain the working mechanism of *SQ-Transformer* and verify our insights in designing these modules.

In summary, *SQ-Transformer* quantizes word embeddings based on their syntactic functions and learns generalizable attention for sentences of the same structure. As a result, it can correctly parse and translate more sentences with unseen compositions of syntactic structures and lexical constituents. We hope this work sheds light on the inner mechanism of Transformers' generalization and inspire future work in architecture design.

2 Background

Vector Quantization (VQ) (Agustsson et al., 2017; Van Den Oord et al., 2017) is a compression technique that represents a set of representations e_x of the variable x by a small, fixed number of code embeddings $\mathbf{z}$. The code is inferred with the nearest neighbor look-up on a codebook $Z \in R^{K \times D}$ made up of K embeddings of the dimension D:

$$q(z_k|x) = \begin{cases} 1 \text{ if } k = \text{argmin}_j \, \text{f}(e_x, z_j) \\ 0 \text{ otherwise} \end{cases} \tag{1}$$
$$\text{VQ}(x) = z_k \text{ where } q(z_k|x) = 1$$

where f is a distance function (e.g., negative cosine similarity).[4] The discrete code embeddings are updated using exponential moving averages of e_x. Previous works (Van Den Oord et al., 2017; Razavi et al., 2019; Ramesh et al., 2021) have shown that VQ-VAE can generate high-fidelity, continuous signals like images and speech. The exploration of VQ on languages (Lingle, 2023) remains limited. In this work, we use VQ to cluster words based on their syntactic function (Sec. 3.1).

Brown Clustering (Brown et al., 1992) is a word clustering algorithm that divides a vocabulary V into m mutually exclusive classes by maximizing the mutual information $I(Z_1, Z_2)$ between the classes of a random bigram (X_1, X_2) in a sentence:

$$\max_{Z:V \to [m]} = \sum_{z_1, z_2} \frac{\#(z_1, z_2)}{N} \log(\frac{\#(z_1, z_2)N}{\#(z_1)\#(z_2)}) \tag{2}$$

where $\#(z, z')$ denotes the number of occurrences of the cluster pair (z, z') for any bigram in $[x_1 ... x_N]$.[5] This algorithm can cluster a vocabulary based on the syntactic functions of words by promoting "predictive clustering": the class of a token must be predictable from the class of its context token. However, it requires nontrivial combinatorial optimization and is difficult to scale and generalize for modern neural networks. In Sec. 3.1.2, we propose a variational Brown Clustering objective that can be optimized with gradient descent.

3 *SQ-Transformer*

In this section, we introduce the components of *SQ-Transformer* and discuss their technical details.

[4] We use the smallest index when there is a tie in argmin.

[5] By assuming a uniform distribution over consecutive word pairs (x_{i-1}, x_i), Brown et al. (1992) approximate $p(z_1, z_2)$ and $p(z)$ using $\frac{\#(z_1, z_2)}{N}$ and $\frac{\#(z)}{N}$ to derive Eqn. 2.

Notations. We denote the source and target sequences as $[x_i]$ and $[y_j]$. The seq2seq framework consists of an encoder with word embeddings E_x and a decoder with word embeddings E_y. For quantizing E_x and E_y, we define two codebooks Z_x and Z_y with K_x and K_y code embeddings respectively.

3.1 Structure-oriented Vector Quantization

Same as the original VQ, Structure-oriented Vector Quantization (SoVQ) clusters the (sub)word embeddings into several classes and quantizes each class into a shared code embedding z (Eqn. 1). We discuss a previous MMI objective and then propose variational Brown Clustering that better cluster words based on their syntactic functions.

3.1.1 Variational MMI objective

Stratos (2019) proposed an unsupervised part-of-speech tagging method by maximizing the mutual information (MMI) between the inferred class Z of a token X and its surrounding context $\hat{X}$. It defines $q(z|x)$ that directly infers the class of x (posterior) and $p(z|\hat{x})$ that predicts the cluster of x based on its context $\hat{x}$ (prior). It maximizes the variational lower bound of the mutual information $I(\hat{X}, Z)$:

$$\begin{aligned} I(\hat{X}, Z) &= H(Z) - H(Z|\hat{X}) \\ &\geq H(Z) - H(q, p) \end{aligned} \tag{3}$$

where $H(q, p)$ is the cross entropy over samples. We show more details about the derivation of this lower bound in Appendix A.1. As we can see in Eqn. 3, maximizing this ELBo is equivalent to (1) minimizing cross-entropy between the cluster inference posterior $q(z|x)$ and cluster prediction prior $p(z|\hat{x})$, so that *words appearing in the same context are assigned to the same class* (we introduce Theorem. 1 to demonstrate this); and (2) maximizing the entropy $H(Z)$ of the cluster distribution to *avoid assigning all words to the same cluster*.

3.1.2 Variational Brown Clustering

In this work, we propose another MMI objective that marries the original Brown Clustering objective $I(Z_1, Z_2)$ and the variational MMI (Eqn. 3). First, we redefine the cluster prediction distribution as $p(z|\hat{z})$, where $\hat{z}$ are the quantized codes of all context tokens $\hat{x}$ inferred from $q(z|\hat{x})$ (Eqn. 1). This differs from the $p(z|\hat{x})$ that predicts the cluster of x directly from its context $\hat{x}$. Then, instead of maximizing the ELBO of $I(\hat{X}, Z)$, we maximize the ELBO of $I(\hat{Z}, Z)$:

$$\begin{aligned} I(\hat{Z}, Z) &= H(Z) - H(Z|\hat{Z}) \\ &\geq H(Z) - H(q(z|x), p(z|\hat{z})) \end{aligned} \tag{4}$$

This inequality is still valid[6] even though we replaced $\hat{X}$ and $p(z|\hat{x})$ in Eqn. 3 with $\hat{Z}$ and $p(z|\hat{z})$. The objective $I(\hat{Z}, Z)$ becomes the exact Brown Clustering objective if we set $\hat{x}$ as a random context token rather than all of them. We show the implementation of this variational loss in Appendix A.4.

We argue that this variational Brown Clustering objective can better cluster words based on their syntactic functions than the lower bound of $I(\hat{X}, Z)$ (Eqn. 3). This is because, according to Theorem. 1, maximizing $I(\hat{X}, Z)$ can only cluster words that appear in similar contexts into the same class. However, some words having the same syntactic function might rarely occur in the same context due to semantics. For example, '*police*' and '*professor*' usually appear in very different contexts: "*The police arrested a thief.*" and "*The professor appraised a student.*" Therefore, maximizing $I(\hat{X}, Z)$ might not push the model to assign them to the same cluster. In comparison, maximizing $I(\hat{Z}, Z)$ (Brown Clustering objective) can encourage clustering structurally equivalent words that appear in various contexts together: even though '*police*' and '*professor*' have different $\hat{X}$, they share the same $\hat{Z}$ given a structure-oriented word cluster.[7] We support this claim with ablations in Sec. 4.

3.2 Systematic Attention Layer

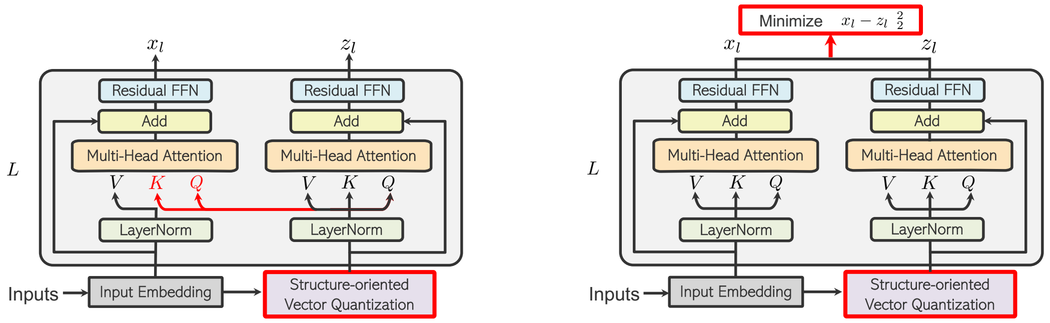
Now that we have quantized each class of words into a code z encoding structural information, we then use z as the queries and keys in computing the attention weights. Here we show the encoder's self-attention module (visualized in Fig. 2a):

$$\begin{aligned} \text{MHAttn}(Q, K, V) &= \text{softmax}(QK^T)V \\ z_{l+1} &= \text{MHAttn}(q = z_l, k = z_l, v = z_l) \\ x_{l+1} &= \text{MHAttn}(q = z_l, k = z_l, v = x_l) \end{aligned}$$

where x_0 and z_0 are the non-contextualized word embeddings and their quantized code embeddings respectively. The two MHAttn modules share all parameters. We call it the Systematic Attention Layer (SAL) because this modified attention module promotes the systematic reusing of attention

[6] We show the derivation in Appendix A.3.

[7] The necessary clustering scheme that can achieve the purpose is [{*arrested,appraised*}, {*thief,student*}].



(a) The Systematic Attention Layer (SAL).

(b) The Systematically Regularized Layer (SRL).

Figure 2: Architecture of the Systematic Attention Layer (SAL) and the Systematically Regularized Layer (SRL). We omit the LayerNorm inside each "residual FFN" module for brevity. Both SAL (a) and SRL (b) run two separate attention streams at training, where SAL (a) additionally shares the key and query between the two streams. In decoding, SRL (b) does not compute the second stream originating from SoVQ.

patterns: as words of the same syntactic function (e.g., '*cat*' and '*dog*', '*asleep*' and '*awake*') are in the same cluster, the transformer would process two sentences of the same syntactic structure ("*The cat is asleep*" and "*The dog is awake*") using the same attention pattern. As a result, a model that understands one sentence is more likely to generalize to the other one, which is the key ability stemming from a systematic language understanding. Similarly, we also use SAL with regular cross attention in the decoder (see Appendix A.5 for details). In summary, SAL enforces **hard attention invariance** among sentences of the same syntactic structure, but at the cost of the flexibility of encoding non-structural relations that commonly exist in natural languages (e.g., idioms, commonsense, etc). We discuss these cases in Sec. 6.

3.3 Systematically Regularized Layer

To encourage systematicity in attention while keeping its ability to encode non-structural relations, we instead use the attention outputs z_l computed from the quantized embeddings to *regularize* the attention outputs x_l computed from word embeddings, by minimizing the squared L2 distances (MSE loss) between z_l and x_l for all layers l:

$$z_{l+1} = \text{MHAttn}(q = z_l, k = z_l, v = z_l)$$
$$x_{l+1} = \text{MHAttn}(q = x_l, k = x_l, v = x_l)$$

where x_0 are the word embeddings. We name it Systematically Regularized Layer (SRL) and visualize it in Fig. 2b. Unlike SAL, SRL demonstrates "**soft invariance**" so that sentences of a common structure are processed with similar (because of the L2 regularization loss) but not necessarily the same attention pattern. During inference, we do not need to perform SoVQ and compute z_l since it is only used to regularize x_l in training. Therefore, SRL does not incur any computation or memory overhead than a vanilla Transformer layer in inference.

4 Experiments

4.1 Datasets

We use three semantic parsing tasks including SCAN ADDJUMP (Lake and Baroni, 2018), AROUNDRIGHT (Loula et al., 2018), COGS (Kim and Linzen, 2020), and the CoGnition (Li et al., 2021) En→Zh translation task, all of which require OOD compositional generalization to test examples. For example, SCAN ADDJUMP tests the models' ability to parse syntactic structures (e.g., "$$x$ twice") combined with a novel entity ($$x$ = '*jump*'), which is never associated with other structures during training. We also use WMT17 English↔German (Bojar et al., 2017) and WMT14 English↔French (Bojar et al., 2014) to test models for generalizability. We introduce each dataset in detail in Appendix B.1.

4.2 Results

Semantic Parsing results. We show the experimental setup in Appendix B.2. We report the results on the two SCAN tasks and COGS in Table 1. Specifically, *SQ-Transformer* with SAL achieves significant[8] improvements over the baseline on SCAN ADDJUMP[9] and AROUNDRIGHT. With SRL, *SQ-Transformer* manages to outperform the

[8] Bootstrapped test with $\alpha < 0.01$.

[9] Both models are trained with 2x augmented SCAN ADDJUMP from Jiang et al. (2022).

Model	JUMP	AROUNDR	COGS
PREVIOUS MODELS			
LSTM-RNN	1.2	2.5±2.7	-
CGPS-RNN	98.8±1.4	83.2±13.2	-
Lex Learn	92.0±0.2	95.0±0	82.0±0
OUR MODELS			
Transformer	40.04±17.3	69.47±9.2	82.60±0.5
SQ-Transformer	**99.42**±1.0*	**99.63**±0.6*	**83.36**±0.7

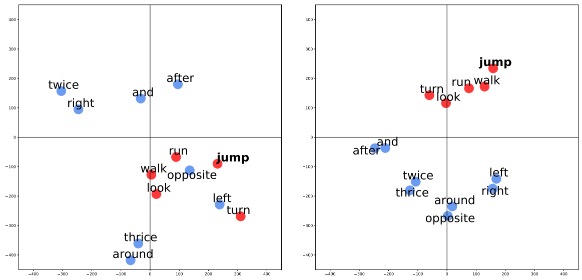
Table 1: Test accuracy from the SCAN ADDJUMP, AROUNDRIGHT, and COGS. We report previous models: LSTM-RNN (Lake and Baroni, 2018), CGPS-RNN (Li et al., 2019), and Lex Learn (Akyurek and Andreas, 2021). We report our models' average (± std.) results from 5 random seeds. *SQ-Transformer* results with * use SAL while others use SRL.

Model	CTER (↓)		BLEU
	Instance	Aggregate	
PREVIOUS MODELS			
Transformer	28.4	62.9	59.5
Proto-Transformer	21.7	51.8	60.1
Dangle-Transformer	22.8	50.6	60.6
Consistency-Reg	20.2	**48.3**	61.3
OUR MODELS			
Transformer	29.55	61.62	60.45
SQ-Transformer$_{\text{SRL}}$	**18.14**	48.89	**62.78**

Table 2: Compound Translation Error Rate (CTER, lower is better) and BLEU on the Compositional Generalization test set from the CoGnition En-Zh. We also report the results from Proto-Transformer (Yin et al., 2022), Dangle-Transformer (Zheng and Lapata, 2022), and consistency-regularized Transformer (Yin et al., 2023). The best result is **bold** and the 2nd is underlined.

Model	En-De	De-En	En-Fr	Fr-En
Transformer	28.10	31.30	37.01	34.24
SQ-Transformer$_{\text{SRL}}$	**29.21**	**31.96**	**38.38**	**35.56**

Table 3: BLEU on WMT17 En↔De, WMT14 En↔Fr.



(a) Source embeddings trained with no quantization.

(b) Source embeddings trained with SoVQ (6 classes).

Figure 3: T-SNE visualization of embeddings learned on SCAN ADDJUMP dataset (Lake and Baroni, 2018).

SQ-Transformer	ADDJUMP	CoGnition
w. SAL	**99.42**±0.98*	59.85±0.49*
w. SRL	47.36±20.83†	**62.35**±0.52†
None	53.79±18.36	61.11±0.34
- SoVQ	78.44±34.01*	60.93±0.13†
- Brown	97.75±3.93*	61.52±0.22†

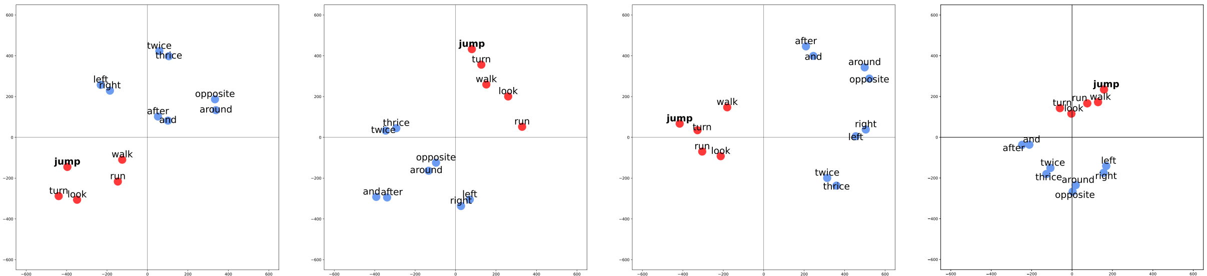
Table 4: Ablation: test accuracy on SCAN ADDJUMP and BLEU on CoGnition (averaged over 5 runs). '- SoVQ' uses the original Vector Quantization. '- Brown' uses the MMI objective from Stratos (2019). Results with * use SAL while results with † use SRL.

baseline on the larger, more natural COGS dataset. This shows the effectiveness of *SQ-Transformer* in generalizing to unseen combinations of syntactic structure and lexical constituents. We compare the performance of SAL and SRL on a small, synthetic dataset and a larger, natural dataset in Sec. 5.1.

Machine Translation results. We evaluate the baseline Transformer as well as *SQ-Transformer* on the CoGnition compositional generalization test set and WMT test sets and report their BLEU 4 (Papineni et al., 2002) scores. For CoGnition, we also report the novel compound translation error (CTER) (Li et al., 2021). It examines whether all of the atoms (tokens) in the novel compound are correctly translated in the generated Chinese sentence. Specifically, instance-level CTER denotes the percentage of the test instances in which one or more atoms in the novel compound are translated incorrectly. Aggregate-level CTER denotes the percentage of novel compounds that are translated wrong in at least one instance. Compared to the Transformer baseline, *SQ-Transformer* obtains significantly higher BLEU scores on CoGnition En→Zh (Table 2), WMT17 En↔De, and WMT14 En↔Fr tasks (Table 3). On CoGnition, *SQ-Transformer* also achieves substantially lower instance and aggregate compound error rate in its Chinese translation. This improvement shows that SoVQ and SRL enable the model to correctly translate more novel compounds: for example, translating "*the dog he liked*" by generalizing from training expressions of the same structure like "*the dress she liked*" and "*an animal she liked*".

5 Analysis

In this section, we conduct an ablation study (Sec. 5.1) and analyze the embeddings (Sec. 5.2) as well as the attention patterns (Sec. 5.3).



(a) n_a=1 (acc=2.08) (b) n_a=10 (acc=4.68) (c) n_a=100 (acc=79.38) (d) n_a=1000 (acc=100.00)

Figure 4: T-SNE visualization of embeddings learned on SCAN ADDJUMP dataset (Lake and Baroni, 2018) with n_a atomic expressions (e.g., *jump*↦JUMP and *walk*↦WALK) for each primitive and the model's accuracy (acc).

5.1 Ablation Study

We conduct an ablation study on the components of *SQ-Transformer* in Table 4. Most notably, we find that SAL only works on small, synthetic datasets like SCAN ADDJUMP but not on CoGnition; while SRL works on larger, more natural datasets like CoGnition and COGS but not on SCAN tasks. This finding corroborates other structure-based attention (Li et al., 2019; Russin et al., 2020) that only achieve strong performance on small, synthetic datasets. It suggests that generalizing on SCAN requires learning its grammar from an extreme data setting, which might only be possible via strictly structural inductive biases like SAL's hard invariance to primitive substitution. Natural languages, on the other hand, require learning both structural and non-structural relations (discussed in Sec. 6), which is not possible using SAL. Later (in Sec. 5.3), we will show how SRL injects a similar, but soft invariance into attention to perform strongly on more natural tasks. Moreover, we show that using the original attention layer ('None') or original VQ ('-SoVQ') results in a significant drop in performance on SCAN ADDJUMP and CoGnition. Finally, optimizing our Variational Brown Clustering loss brings extra benefits compared to the MMI ('-Brown') objective Stratos (2019).

5.2 Analyzing the Embedding Space

Visualizing SCAN embedding space. We visualize the embedding matrices using 2-d t-distributed Stochastic Neighbor Embedding (t-SNE) , which projects each embedding into a 2-dimensional coordinate (Hinton and Roweis, 2002). In Fig. 3b, we show that the source embeddings learned by SoVQ on SCAN ADDJUMP are clustered based on their syntactic functions in a sentence structure: the conjunction words ('*and*', '*after*'), direction adverbs ('*left*', '*right*'), prepositions ('*around*', '*opposite*'), and adverbs ('*twice*', '*thrice*') are clustered together respectively. Most importantly, the rare primitive '*jump*' is clustered together with other primitives. This enables the *SQ-Transformer* with SAL to generalize to unseen expressions like "*jump twice*" by reusing the same attention pattern as it has learned for "*walk twice*". On the contrary, in Fig. 3a, words of the same syntactic functions (e.g., '*jump*' and '*walk*') are distant apart in the t-SNE space. This prevents the Transformer from generalizing to novel expressions like "*jump twice*". Finally, SoVQ also learns structure-based word clusters on COGS that has a larger vocab (discussed in Appendix C.2).

Case Study: Learning the syntactic equivalence of 'jump' and 'walk'. The major challenge in SCAN ADDJUMP is to recognize the equivalent syntactic function of the rare primitive '*jump*' and other common primitives like '*walk*' based on the only syntactic structure[10] that has both '*jump*' and '*walk*' as a constituent. Next, we present a case study of how SoVQ manages to learn this equivalence and demonstrate three necessary preconditions. **First, it is important to choose the proper number of clusters so that the model cannot afford to reserve a cluster for '*jump*' only.** For example, if we initialize 12 classes instead of 6 classes for the vector quantization, the model will put '*jump*' into a separate class from '*walk*' and '*run*'. **Second, the model must be exposed to a sufficient number of examples that put '*jump*' and '*walk*' in the same syntactic structure (context).** In Fig. 4, we show the t-SNE visualization of the source embeddings and generalization accuracy when the model is trained with different (n_a) repe-

[10] Atomic expressions like "*jump*↦JUMP", "*walk*↦WALK."

titions of atomic expressions for each primitive.[11] In all four cases, SoVQ can roughly cluster words based on their syntactic functions (e.g., 'twice' and 'thrice' are always together). However, there are some subtle differences regarding the clustering of verb primitives (shown in red). When n_a equals 1 or 10 (Fig. 4a, Fig. 4b), *SQ-Transformer* can hardly generalize (accuracy < 5%) and the '*jump*' is located distantly with other verbs in red. With 100/1000 atomic expressions per primitive (Fig. 4c, Fig. 4d), *SQ-Transformer* achieves significantly improved generalization and learns more compact clusters of verbs. **Finally, our variational Brown Clustering objective is indispensable for clustering words based on their syntactic functions.** This can be seen in both final results (Table 4) and the distribution of the embedding space (Fig. 6).

5.3 Analyzing the Attention Patterns

In this part, we reveal that *SQ-Transformer* indeed learns attention patterns that systematically generalize to novel expressions. As we have shown above, SoVQ can cluster words in the SCAN vocab into multiple structural equivalence classes $\{\mathcal{C}_i\}$. Therefore, by computing the queries and keys using the quantized embeddings, Systematic Attention Layers (SAL) are guaranteed to produce the same attention maps given any examples of a common structure (e.g., $\$x$ *around left* $\forall \$x \in \mathcal{C}_i$).

Next, we focus on analyzing the attention maps learned by the Systematically Regularized Layer (SRL). Unlike SAL, SRL still computes the attention weights using word embeddings as the queries and keys, but is regularized to keep its outputs close to the code representations from SAL. To evaluate the effectiveness of regularization, we collect 48 pairs of source sentences from the CoGnition test set. The sentences within each pair are quantized into the same sequence of codes (e.g., "*he stopped every girl.*" and "*he found each child.*") by SoQV.[12] We then count the percentage of attention heads that assign the highest weight to the same token when processing two examples in a pair. In this metric, *SQ-Transformer* achieves 79.8% compared to the baseline with 72.8%. This demonstrates that SRL learns systematic attention patterns in representing structures, while maintaining generalizability to natural data like CoGnition and WMT.

6 Discussion

In this part, we (1) discuss why the Systematic Attention Layer (SAL) can only work on small, synthetic datasets like SCAN; and (2) further motivate the Systematically Regularized Layer (SRL) that achieves a balance between compositionally encoding structures and maintaining flexibility in incorporating non-compositional relationships that commonly exist in natural languages.

Recall SAL represents the "common structural relations" by computing attention weights in the quantized, syntactic space. This ensures that sentences that have (1) the same structure and (2) equivalently cognizable entities at all positions[13] are processed with the same attention weights across all heads and layers. However, natural languages are only approximately compositional, indicating that sometimes the meaning of a piece of text does not only depend on its structure and its lexical constituents. We discuss two situations where SAL is overly strict and thus prevents the model from encoding non-structural linguistic features.

Situation 1: Semantics. We use the classic Winograd Challenge (Levesque et al., 2012) to demonstrate how lexical semantics, or more specifically, commonsense knowledge affects the comprehension of a sentence. Here, the model sees two sentences that differ only in one or two words, which are often of the same syntactic role but contain a referential ambiguity resolved in opposite directions. For example:

- The trophy doesn't fit in the brown suitcase because it's too big. What is too big? **Answer**: The trophy.
- The trophy doesn't fit in the brown suitcase because it's too small. What is too small? **Answer**: The suitcase.

Here, it is necessary to use commonsense semantics to resolve the ambiguous anaphora. However, the attention maps learned by SAL for these two sentences would be the same, given that "big" and "small" are most likely quantized into the same class. Therefore, a model with SAL cannot encode this commonsense knowledge into the attention.

Situation 2: Pragmatics. Other than the inability to incorporate commonsense to resolve the coreference within a sentence, SAL with "hard invariance" over structural equivalence classes is only

[11] "n_a=1000" means there are 1000 "*jump*↦`JUMP`", 1000 "*walk*↦`WALK`", etc. in the training set.

[12] We show all 48 pairs of sentences in Table 6.

[13] E.g., "*The cat is awake*" and "*The dog is asleep*".

designed for capturing local dependency within a sentence. It cannot scale to represent the complex relationships between words across sentences. Previously, Sartran et al. (2022) bias the attention with the sentence parse tree, and reported deteriorated performance on document-level language modeling. Similarly, we observe that using quantized embeddings to compute cross-attention weights would result in degenerated performance on SCAN tasks, and thus opt for the original word embeddings as the queries and keys (Appendix A.5).

In both situations, it is necessary to compute the attention using the word embeddings rather than their quantized code embedding, so that the attention can capture other non-structural relations (e.g., commonsense) as well. Empirically, we also show that using the SAL results in degenerated performance on COGS and CoGnition datasets (Table 4), which include a much larger vocab plus more natural and longer expressions. Therefore, as we motivated in Sec. 3.3, we instead use the more flexible SRL to achieve strong performance on these tasks.

7 Other Related Work

Compositional generalization. Earlier works investigated compositionality of neural networks in language learning (Wong and Wang, 2007; Brakel and Frank, 2009), compositional counting (Wiles, 1998; Weiss et al., 2018), and syntax learning (Linzen et al., 2016). Recent works (Lake and Baroni, 2018; Kim and Linzen, 2020; Loula et al., 2018; Bastings et al., 2018; Keysers et al., 2020; Tsarkov et al., 2021; Hupkes et al., 2020) embed the compositional challenge into semantic parsing tasks and directly evaluate seq2seq models on an out-of-distribution test set.

Previous works have also proposed many novel methods to improve the compositional generalization of neural models. Such methods include novel architectures (Li et al., 2019; Russin et al., 2020; Dessì and Baroni, 2019; Gordon et al., 2020; Oren et al., 2020; Zheng and Lapata, 2021), grammar-based approaches (Shaw et al., 2021; Kim, 2021), task decomposition (Herzig et al., 2021), data augmentation (Andreas, 2020; Akyürek et al., 2021; Akyürek and Andreas, 2022), careful architecture selection (Csordás et al., 2021), and novel learning methods (Lake, 2019; Conklin et al., 2021; Jiang et al., 2022; Xu et al., 2022).

Structures captured by Transformer. Researchers have long been studying how the attention maps of Transformers encode the syntactic structure (e.g., dependency parse) of a sentence (Hewitt and Manning, 2019; Phang et al., 2019; Clark et al., 2019; Limisiewicz et al., 2020; Murty et al., 2023a). Recently, Jian and Reddy (2023) substituted words in a sentence with words from the same syntactic category and then averaged the attention maps of a BERT (Devlin et al., 2019) model across these "syntactically invariant sentences". Our analysis in Fig. 1 also makes use of such syntactically invariant sentences. We further reveal the correlation between the emergence of systematic attention patterns and the model's generalization ability in a Transformer trained from scratch, which leads to our effort to inject linguistic structure into the model, as discussed below.

Incorporating linguistic knowledge into models. Many previous works have tried to incorporate linguistically-informed labels, especially syntactic labels, into neural networks (Sennrich and Haddow, 2016; Strubell et al., 2018; Sachan et al., 2021; Qian et al., 2021; Sartran et al., 2022). Following this idea, later works explored the syntactic equivalence of '*jump*' and other verbs to induce compositionality (Akyurek and Andreas, 2021; Jiang and Bansal, 2021; White and Cotterell, 2022) for SCAN. Most relevantly, Chakravarthy et al. (2022) manually implemented several abstract "roles" for tokens in SCAN and computed the attention using the role embeddings. Our work differs from most of these works in that we do not require any external labels (e.g., parse tree) of sentences and instead automatically infer the syntactic "roles" for each token using Structure-oriented Vector Quantization and leverage them in Systematic Attention Layers and Systematically Regularized Layers.

8 Conclusion

In this work, we propose *SQ-Transformer* with Structure-oriented Vector Quantization and two types of attention layers that use the quantized embeddings as the keys and values. Our experiments show that *SQ-Transformer* can better generalize to unseen expressions in multiple semantic parsing and machine translation tasks. We conduct multiple analyses and show that SoVQ can cluster word embeddings based on their syntactic roles and the model learns systematic attention patterns in processing sentences of the same syntactic structure.

9 Limitations

In this work, the proposed Structure-oriented Vector Quantization (SoVQ) mainly clusters the uncontextualized, *lexical* constituents based on their syntactic roles. It does not explicitly encourage *phrasal* constituents with the same syntactic role to be close together in the representation space. Therefore, we find that *SQ-Transformer* does not achieve better performance than the vanilla Transformer on COGS test examples with "deeper recursion depth" or "novel combination modified phrases and grammatical roles", both of which require generalizing to novel combinations of grammatical structures and phrasal constituents.

SoVQ on uncontextualized embeddings also does not consider polysemy, which requires assigning a token to potentially different syntactic classes based on its context (discussed in Appendix D.1). However, our work is the first to apply vector quantization to improve compositional generalization of Transformers. We showed that clustering the lexical embeddings can already improve the Transformer in learning a compositional representation of the sentences. Therefore, we believe releasing our promising results by structurally clustering lexical components can potentially inspire more researchers in the community to take on this challenging but rewarding task of structurally clustering contextualized, phrasal representations.

Same as other Vector Quantization methods, we need to manually set the number of classes for clustering/quantizing embeddings (hyperparameter search details discussed in Appendix B.2). To prove the effectiveness of SoVQ, we initialize 4 clusters for SCAN tasks (whose vocabulary size is smaller than 20) and 16/32 clusters for tasks like CoGnition and WMT. This is because having too many clusters loosens the information bottleneck and fails to induce generalization. Future work incorporating SoVQ into the pretraining stage of foundation models does not need to tune the number of classes for every new task, but only needs to set it once given the size of its universal vocabulary.

10 Ethics Statement

Our *SQ-Transformer* architecture is designed to build models with a stronger ability to generalize to unseen compositions. It can be used as the backbone of a large foundation model with better data efficiency in acquiring a generalizable understanding of natural languages. However, we note the models to be built with our architecture should be used with careful consideration. Since this is only a study aiming to improve the generalization ability of Transformer models and we do not release or plan to release a model pretrained for general use, we thus do not study the undesired behavior of the proposed model. Therefore, future works, which intend to use our SQ-Transformer or any proposed components (SoVQ, Systematic Attention Layer, and Systematically Regularized Layer) in training a large language model for real-world usage, need to conduct a thorough study on its safety and potential risks (similar to usage of any other deep learning models), including but not limited to honesty, truthfulness, fairness, toxicity, etc.

Acknowledgements

We thank Elias Stengel-Eskin, Peter Hase, Archiki Prasad, Swarnadeep Saha, and Yi-Lin Sung for their useful feedback. This work was supported by NSF-CAREER Award 1846185, DARPA MCS Grant N66001-19-2-4031, and an Apple Scholars in AI/ML PhD Fellowship. The views are those of the authors and not of the funding agency.

References

Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. 2017. Soft-to-hard vector quantization for end-to-end learning compressible representations. *Advances in neural information processing systems*, 30.

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *International Conference on Learning Representations*.

Ekin Akyurek and Jacob Andreas. 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.

Ekin Akyürek and Jacob Andreas. 2022. Compositionality as lexical symmetry. *arXiv preprint arXiv:2201.12926*.

Jacob Andreas. 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.

Jasmijn Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. 2018. Jump to better conclusions: SCAN both left and right. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 47–55, Brussels, Belgium. Association for Computational Linguistics.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Philémon Brakel and Stefan Frank. 2009. Strong systematicity in sentence processing by simple recurrent networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480.

Paco Calvo and John Symons. 2014. *The Architecture of Cognition: Rethinking Fodor and Pylyshyn's Systematicity Challenge*. MIT Press.

Ayush K Chakravarthy, Jacob Labe Russin, and Randall O'Reilly. 2022. Systematicity emerges in transformers when abstract grammatical roles guide attention. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 1–8, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Noam Chomsky. 1957. *Syntactic structures*. De Gruyter Mouton.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. 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 3322–3335, Online. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roberto Dessì and Marco Baroni. 2019. CNNs found to jump around more skillfully than RNNs: Compositional generalization in seq2seq convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3919–3923, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2023. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

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

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. *arXiv preprint arXiv:2104.07478*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15.

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

Jasper Jian and Siva Reddy. 2023. Syntactic substitutability as unsupervised dependency syntax. In *Proceedings of EMNLP*.

Yichen Jiang and Mohit Bansal. 2021. Inducing transformer's compositional generalization ability via auxiliary sequence prediction tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6253–6265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yichen Jiang, Xiang Zhou, and Mohit Bansal. 2022. Mutual exclusivity training and primitive augmentation to induce compositionality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11778–11793, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kent Johnson. 2004. On the systematicity of language and thought. *The Journal of Philosophy*, 101(3):111–139.

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

Najoung Kim and Tal Linzen. 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.

Yoon Kim. 2021. Sequence-to-sequence learning with latent neural grammars. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 26302–26317.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. 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 4767–4780, Online. Association for Computational Linguistics.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 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.

Tomasz Limisiewicz, David Mareček, and Rudolf Rosa. 2020. Universal Dependencies According to BERT: Both More Specific and More General. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2710–2722, Online. Association for Computational Linguistics.

Lucas D Lingle. 2023. Transformer-vq: Linear-time transformers via vector quantization. *arXiv preprint arXiv:2309.16354*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

João Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium. Association for Computational Linguistics.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Gary F Marcus. 2003. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

David McAllester. 2018. Information theoretic co-training. *arXiv preprint arXiv:1802.07572*.

Richard Montague. 1970. Universal grammar. *1974*, pages 222–46.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. 2023a. Characterizing intrinsic compositionality in transformers with tree projections. In *The Eleventh International Conference on Learning Representations*.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. 2023b. Pushdown layers: Encoding recursive structure in transformer language models. *arXiv preprint arXiv:2310.19089*.

Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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.

Jason Phang, Shikha Bordia, Samuel R Bowman, et al. 2019. Do attention heads in bert track syntactic dependencies? In *NY Academy of Sciences NLP, Dialog, and Speech Workshop*.

Steven Phillips and William H Wilson. 2016. Systematicity and a categorical theory of cognitive architecture: universal construction in context. *Frontiers in psychology*, 7:1139.

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural guidance for transformer language models. 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 3735–3745, Online. Association for Computational Linguistics.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

Jacob L Russin, Jason Jo, Randall C O'Reilly, and Yoshua Bengio. 2020. Systematicity in a recurrent neural network by factorizing syntax and semantics. In *CogSci*.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? 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 922–938, Online. Association for Computational Linguistics.

Karl Stratos. 2019. Mutual information maximization for simple and accurate part-of-speech induction. 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 1095–1104, Minneapolis, Minnesota. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE.

Dmitry Tsarkov, Tibor Tihon, Nathan Scales, Nikola Momchev, Danila Sinopalnikov, and Nathanael Schärli. 2021. *-cfq: Analyzing the scalability of machine learning on a compositional task. In *AAAI*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Jennifer C. White and Ryan Cotterell. 2022. Equivariant transduction through invariant alignment. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4651–4663, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Paul Rodriguez Janet Wiles. 1998. Recurrent neural networks can learn to implement symbolsensitive counting. *Advances in Neural Information Processing Systems*, 10:87.

Francis CK Wong and William SY Wang. 2007. Generalisation towards combinatorial productivity in language acquisition by simple recurrent networks. In *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 139–144. IEEE.

Zhenlin Xu, Marc Niethammer, and Colin A Raffel. 2022. Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. *Advances in Neural Information Processing Systems*, 35:25074–25087.

Yongjing Yin, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2022. Categorizing semantic representations for neural machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5227–5239, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308, Toronto, Canada. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2021. Compositional generalization via semantic tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1022–1032, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2022. Disentangled sequence to sequence learning for compositional generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4256–4268, Dublin, Ireland. Association for Computational Linguistics.

Xiang Zhou, Yichen Jiang, and Mohit Bansal. 2023. Data factors for better compositional generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Appendix

A Structure-oriented Vector Quantization

Notations. We denote the source sequence as $[x_1...x_N]$ and the target sequence as $[y_1...y_M]$. The framework consists of an encoder with word embeddings $E_x \in \mathbb{R}^{N_x \times D}$ and an autoregressive decoder with word embeddings $E_y \in \mathbb{R}^{N_y \times D}$, where N_x and N_y are the number of tokens in vocabulary and D is the dimension of the embeddings. For quantizing E_x and E_y, we define two codebooks $Z_x \in \mathbb{R}^{N_x \times D_z}$ and $Z_y \in \mathbb{R}^{N_y \times D_z}$, where N_x and N_y are the number of codes and D_z is the dimension of the code embedding.

A.1 Generalized Brown Clustering Objective.

Here, we present the details of the Brown Clustering (Brown et al., 1992)., the generalized Brown Clustering objective (Stratos, 2019), and its Evidence Lower Bound (ELBO) Eqn. 3 (with proofs) discussed in Sec. 3.1.

Brown Clustering. Brown Clustering is an unsupervised word clustering algorithm that was proposed and popularized before the trend of neural networks. Brown clustering divides a vocabulary V into m clusters by maximizing the mutual information (MMI) between the clusters (Z_X, Z_Y) of a random bigram (X, Y) in a corpus of N words $(x_1...x_N)$. The algorithm assumes a uniform distribution over consecutive word pairs (x_{i-1}, x_i) and optimizes the MMI objective:

$$\max_{Z:V\to[m]} = \sum_{z,z'} \frac{\#(z,z')}{N} \log(\frac{\#(z,z')N}{\#(z)\#(z)}) \quad (5)$$

where $\#(z, z')$ denotes the number of occurrences of the cluster pair (z, z') for any bigram in $(x_1...x_N)$. This objective is intractable, so (Brown et al., 1992) proposed a greedy algorithm that (1) initializes the clusters by assigning each word to a distinct class and then (2) merges the pair of classes that leads to the least loss in the average mutual information for a total of $|V| - m$ times.

This original algorithm failed on vocabularies larger than 5000 words. To the remedy, the authors instead (1) initialize m classes that each contain one of the m most frequent words and (2) repeatedly merge a pair of clusters that yields the smallest decrease in mutual information. However, this heuristic requires nontrivial combinatorial optimization and is difficult to scale and generalize.

The Generalized MMI Objective. Stratos (2019) generalized the Brown Clustering to the setting by maximizing the mutual information between the posterior clustering probability $q(z|x)$ and prior $p(z|\hat{x})$, where x is a random token from a sentence and $\hat{x}$ is its surrounding context. Since p and q are conditionally independent given $(x, \hat{x})$, we have $p(z, z'|x, \hat{x}) = q(z|x)p(z'|\hat{x})$. Thus, the mutual information between p and q given a single sentence $(x, \hat{x})$ is:

$$J_{x,\hat{x}} = \sum_{z,z'} q(z|x)p(z'|\hat{x})\log\frac{q(z|x)p(z'|\hat{x})}{q(z)p(z')} \quad (6)$$

The author then maximizes $\mathbb{E}_{x,\hat{x}\sim D}[J_{x,\hat{x}}]$. This objective becomes the exact Brown Clustering in Eqn. 5 if (1) $\hat{x}$ is a random context token; (2) p and q are tied and are hard clustering instead of probabilistic soft clustering.

A.2 A Variational Lower Bound of MMI.

Stratos (2019) further improved the generalized Brown Clustering and derived a novel objective as the difference between the entropy of the cluster distribution $H(Z)$ and the cross entropy between q and p:

$$J^{var} = H(Z) - H(q, p) \quad (7)$$

where $H(q, p)$ is the cross entropy between q and p over samples:

$$H(q, p) = \mathop{\mathbb{E}}_{x,\hat{x}\sim D}\left[-\sum_z q(z|x)\log p(z|\hat{x})\right] \quad (8)$$

Intuitively, minimizing the cross entropy between q and p can improve their mutual information. Maximizing $H(Z)$ encourages the equal occurrence of each cluster and can prevent the trivial solution that assigns all tokens x in the corpus into the same cluster.

The objective in Eqn. 7 can also be seen as the lower bound of the mutual information between two random variables $I(\hat{X}, Z)$, where $\hat{X}$ is the context of a token X and Z is its cluster inferred from $q(z|x)$. This lower bound is shown in McAllester (2018) and we replicate it below.

First, since $I(\hat{X}, Z) = H(Z) - H(Z|\hat{X})$ and by definition we have the conditional entropy:

$$\begin{aligned} &H(Z|\hat{X}) \\ &= -\sum_{\hat{x},z} p(\hat{x}, z)\log\frac{p(\hat{x}, z)}{p(\hat{x})} \\ &= \sum_{\hat{x},z} p(\hat{x}, z)\log\frac{p(\hat{x})}{p(\hat{x}, z)} \\ &= \sum_{\hat{x},z} p(\hat{x}, z)\log\frac{1}{\pi(z|\hat{x})} \\ &= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D \\ z\sim q(\cdot|x)}}[\log\frac{1}{\pi(z|\hat{x})}] \end{aligned} \quad (9)$$

where $\pi(z|\hat{x})$ is the ground-truth prior probability of the cluster of a token given its context. If we introduce a variational distribution $p(z|\hat{x})$ to approximate $\pi(z|\hat{x})$ and further expand Eqn. 8, we have

$$\begin{aligned} &H(q, p) \\ &= \mathop{\mathbb{E}}_{(x,\hat{x})\sim D}\left[-\sum_z q(z|x)\log p(z|\hat{x})\right] \\ &= \mathop{\mathbb{E}}_{(x,\hat{x})\sim D}\left[\sum_z q(z|x)\log\frac{1}{p(z|\hat{x}})\right] \\ &= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D \\ z\sim q(\cdot|x)}}\left[\log\frac{1}{p(z|\hat{x})}\right] \\ &= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D \\ z\sim q(\cdot|x)}}\left[\log\frac{\pi(z|\hat{x})}{\pi(z|\hat{x})p(z|\hat{x})}\right] \\ &= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D \\ z\sim q(\cdot|x)}}\left[\log\frac{1}{\pi(z|\hat{x})}\right] + \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D \\ z\sim q(\cdot|x)}}\left[\log\frac{\pi(z|\hat{x})}{p(z|\hat{x})}\right] \\ &= H(Z|\hat{X}) + D_{\mathrm{KL}}(\pi||p) \end{aligned} \quad (10)$$

where D_{KL} is Kullback–Leibler divergence. Therefore, we can see that $H(p, q)$ is an upper bound of $H(Z|\hat{X})$, and hence $H(z) - H(q, p)$ is the lower bound of $I(\hat{X}, Z)$. Stratos (2019) argues that maximizing $I(\hat{X}, Z)$ over the cluster inference distribution q enforces "predictive coding" because it pushes the cluster inferred from $q(z|X)$ to be more informative of the context $\hat{X}$.

As we can see in Eqn. 3, maximizing this ELBo is equivalent to (1) minimizing cross-entropy between the cluster inference posterior $q(z|x)$ and cluster prediction prior $p(z|\hat{x})$ and (2) maximizing the entropy $H(Z)$ of the cluster distribution.

First, minimizing the cross-entropy $H(q,p)$ enforces "predictive clustering": the class of x must be predictable from its context $\hat{x}$. We introduce a theorem to show how this leads to assigning words appearing in the same context to the same class.

Theorem 1. *Let x_a and x_b be two tokens that only appear in the same sets of context $\hat{X}$. Let $p', q' =$*

$$\arg\min_{p,q} H(q(z|x_a), p(z|\hat{x})) + H(q(z|x_b), p(z|\hat{x}))$$

Then, we have: $q'(z|x_a) = q'(z|x_b) \quad \forall z \in Z$, which means x_a and x_b are clustered into the same class in the optimal solution.

The proof is straightforward: the minimum value of the cross-entropy loss is 0, and this can only be achieved when $q(z|x_a) = q(z|x_b) = p(z|\hat{x})$. To better understand the mathematical intuition of this theorem, consider the case where all adjectives are categorized into the same cluster. Then, we can confidently predict the class of $\$x$='*amazing*' based on the context "*The food tastes* $\$x$.", thus achieving the low cross-entropy with the posterior.

Second, maximizing the entropy of the cluster distribution pushes the model to utilize every cluster z in the latent space with (almost) equal probability. It thus prevents the trivial solution that assigns all tokens to only one random cluster k: $p(z_k|x) = q(z_k|x) = 1$ to minimize the first cross-entropy term ($H(p,q) = 0$). Empirically, this variational MMI objective achieves strong unsupervised POS tagging performance (Stratos, 2019).

A.3 The Variational Brown Clustering

In this work, we propose another MMI objective that marries the original Brown Clustering objective (Eqn. 5) and the alternative objective (Eqn. 3). First, we redefine the cluster prediction distribution as $p(z|\hat{z})$, where $\hat{z}$ are the quantized codes of all context tokens $\hat{x}$ inferred from $\text{argmax}(q(z|\hat{x}))$. This differs from the $p(z|\hat{x})$ that predicts the cluster of x directly from its context $\hat{x}$. Then, instead of maximizing the ELBO of $I(\hat{X}, Z)$, we maximize the ELBO of $I(\hat{Z}, Z)$. Next, we derive the lower bound of $I(\hat{Z}, Z)$ in a similar way to Eqn. 9 and Eqn. 10. First, by definition we have:

$$I(\hat{Z}, Z) = H(Z) - H(Z|\hat{Z})$$

Then, we rewrite $H(Z|\hat{Z})$ as:

$$\begin{aligned}
&H(Z|\hat{Z})\\
&= -\sum_{\hat{z},z} p(\hat{z},z)\log\frac{p(\hat{z},z)}{p(\hat{z})}\\
&= \sum_{\hat{z},z} p(\hat{z},z)\log\frac{p(\hat{z})}{p(\hat{z},z)}\\
&= \sum_{\hat{z},z} p(\hat{z},z)\log\frac{1}{\pi(z|\hat{z})} \qquad (11)\\
&= \sum_{\hat{z},z}\sum_{\hat{x},x} p(\hat{x},x)p(\hat{z},z|\hat{x},x)\log\frac{1}{\pi(z|\hat{z})}\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ z\sim q(\cdot|x)\\ \hat{z}\sim q(\cdot|\hat{x})}}[\log\frac{1}{\pi(z|\hat{z})}]
\end{aligned}$$

where $\pi(z|\hat{z})$ is the ground-truth code prediction distribution given its context's codes. If we introduce a variational distribution $p(z|\hat{z})$ and rewrite Eqn. 8 with the newly defined $p(z|\hat{z})$, we have:

$$\begin{aligned}
&H(q,p)\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})}}\left[-\sum_z q(z|x)\log p(z|\hat{z})\right]\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})}}\left[\sum_z q(z|x)\log\frac{1}{p(z|\hat{x})}\right]\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})\\ z\sim q(\cdot|x)}}\left[\log\frac{1}{p(z|\hat{z})}\right]\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})\\ z\sim q(\cdot|x)}}\left[\log\frac{\pi(z|\hat{z})}{\pi(z|\hat{x})p(z|\hat{z})}\right]\\
&= \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})\\ z\sim q(\cdot|x)}}\left[\log\frac{1}{\pi(z|\hat{z})}\right] + \mathop{\mathbb{E}}_{\substack{(\hat{x},x)\sim D\\ \hat{z}\sim q(\cdot|\hat{x})\\ z\sim q(\cdot|x)}}\left[\log\frac{\pi(z|\hat{z})}{p(z|\hat{z})}\right]\\
&= H(Z|\hat{Z}) + D_{\text{KL}}(\pi||p)
\end{aligned}$$

Therefore, $H(q,p)$ is the upper bound of $H(Z|\hat{Z})$, and hence $H(Z) - H(q,p)$ is the lower bound of $I(\hat{Z}, Z)$.

A.4 Implementation of MMI objectives

Finally, we show our empirical implementation of the MMI objective. First, we implement $q(z|x)$ as the cosine similarity between the word embedding

e_x and code embedding z. We implement $p(z|\hat{x})$ as a separate Transformer network that takes the context[14] of x as the input and predicts the class of x. Following Stratos (2019), we estimate these terms from the training data:

$$\begin{aligned} q'(z) &= \frac{1}{N}\sum_{i=1}^{N} q(z|x_i) \\ H'(Z) &= -\sum_{z} q'(z)\text{log}q'(z) \\ H'(p,q) &= \frac{1}{N}\sum_{i=1}^{N}(-\sum_{z} q(z|x_i)\text{log}p(z|\hat{z}_i)) \end{aligned} \tag{12}$$

where N is the number of tokens. We then add $\alpha(H'(p,q) - H'(Z))$ to the overall loss, where α is a tunable coefficient.

A.5 Systematic Attention Layer for Decoder

In Sec. 3.2, we introduce the Systematic Attention Layer (SAL) for a Transformer encoder. Here, we introduce SAL for a Transformer decoder with encoder-decoder cross attention. we can use the quantized embedding z as the queries and keys in computing the self-attention weights but use the word embedding x as the queries and keys for cross attention. This is because we find that long-term, cross-sentence relationships cannot be determined by words' syntactic functions only (Sec. 6):

$$\begin{aligned} z_l &+= \text{SelfAttn}(q = z_{l-1}, k = z_{l-1}, v = z_{l-1}) \\ z_l &+= \text{CrossAttn}(q = z_l, k = z_L^x, v = z_L^x) \\ y_l &+= \text{SelfAttn}(q = z_{l-1}, k = z_{l-1}, v = y_{l-1}) \\ y_l &+= \text{CrossAttn}(q = y_l, k = x_L, v = x_L) \end{aligned}$$

where y_0 and z_0 are the non-contextualized word embeddings and their quantized code embeddings respectively, x_L and z_L^x are the encoder's final layer outputs (as shown in Fig. 2a). In the equation above, the cross-attention that computes y_l (last row) is the same as a regular Transformer layer. After the final layer, we project z_L and supervise it to predict the code of the next token, same as how we project x_L to predict the next token.

A.6 Information Bottleneck Interpretation

Tishby and Zaslavsky (2015) state that to have a generalizable deep neural network, it is necessary to optimize the Information Bottleneck (IB) tradeoff between compression and prediction for each layer. It is equivalent to minimizing the Lagrangian $I(X_{l-1}, X_l) - \beta I(X_l, Y)$, where X_l is a mapping (e.g., representation produced by the l-th layer in a neural net) of X and $I(X_l, Y)$ is the mutual information between X_l and the label Y. This objective suggests that we need to find the most concise representation X_l that is also sufficient to encode $I(X_l, Y)$ at each layer l. On the one hand, minimizing $I(X_{l-1}, X_l)$ can prevent redundant information from flowing to the next layer, which could be exploited to establish some spurious correlations between X_l and Y. On the other hand, maximizing $I(X_l, Y)$ ensures that sufficient information is encoded in layer l to enables the final prediction of Y.

We argue that SoVQ and SRL can implicitly minimize $I(X_{l-1}, X_l)$ for $l = 0, 1...L$: (1) SoVQ clusters the N word embeddings X_0 around K code embeddings ($N > Z$[15]), thus achieving a lower $H(X_0)$ compared to an unrestricted embedding space and minimizing $I(X, X_0)$; (2) SRL, on the other hand, clusters the layer's outputs computed from word embeddings around the layer's outputs computed from code embeddings. It thus reduces $H(X_l)$ to minimize $I(X_{l-1}, X_l)$ for $l = 1...L$. Therefore, from the standpoint of information theory, SoVQ and SRL impose information bottlenecks on the embedding layer and every attention layer to improve the generalization of the entire network.

B Experiments

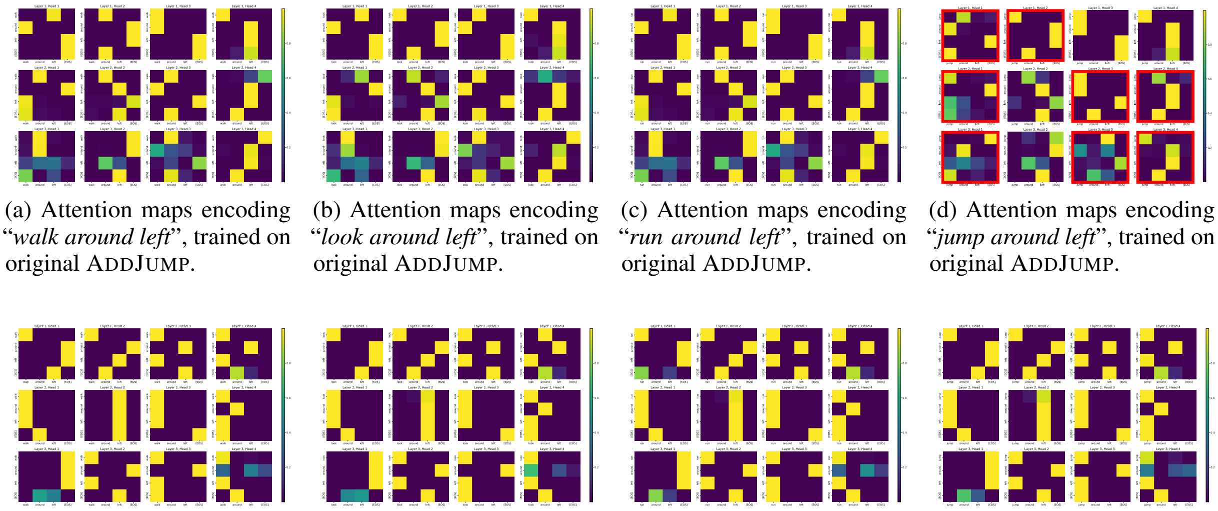
B.1 Datasets

We use a series of semantic parsing and machine translation tasks requiring compositional generalization and the common WMT tasks. All semantic parsing datasets are in English. None of the datasets includes any information that can name or uniquely identify individual people or offensive content. Since SCAN, COGS, and CoGnition do not have a compositional generalization validation set, we randomly sample 20% of the test data and use it as the validation set.

SCAN ADDJUMP (Lake and Baroni, 2018) tests the models' ability to generalize syntactic structures (e.g., "$x twice") to a novel entity ($x =

[14] The context can be either the whole sentence with x masked (bidirectional context) or the preceding words of x only (left context).

[15] For example, we only use 16 codes (K=16) to quantize 40356 subword embeddings for the WMT17 En-De tasks.



(a) Attention maps encoding "*walk around left*", trained on original ADDJUMP.

(b) Attention maps encoding "*look around left*", trained on original ADDJUMP.

(c) Attention maps encoding "*run around left*", trained on original ADDJUMP.

(d) Attention maps encoding "*jump around left*", trained on original ADDJUMP.

(e) Attention maps encoding "*walk around left*", trained on 20x augmented ADDJUMP.

(f) Attention maps encoding "*look around left*", trained on 20x augmented ADDJUMP.

(g) Attention maps encoding "*run around left*", trained on 20x augmented ADDJUMP.

(h) Attention maps encoding "*jump around left*", trained on 20x augmented ADDJUMP.

Figure 5: Attention maps encoding training examples "*walk around left*", "*look around left*", "*run around left*" and a test example "*jump around left*" from the Transformers trained on the original SCAN ADDJUMP training set (a, b, c, d), and 20x augmented training set (Zhou et al., 2023) (e, f, g, h). We highlight the attention patterns in (d) that differ from (a), (b), (c) in red boxes. When trained on 20x augmented training set, the model encodes the two examples (c and d) with highly similar attention maps across all layers and heads.

'*jump*'). Here we use the augmented training set with 2 times more primitives (Jiang et al., 2022), which includes 97906 examples. The validation and test sets have 1541 and 7706 examples respectively.

SCAN AROUNDRIGHT (Loula et al., 2018) tests the models' ability to generalize a common syntactic structure "$\$x_1$ around $\$x_2$" to an entity ($\$x_2$='*right*') that is only associated with other structures during the training. The train, validation, and test sets have 15225, 895, and 4476 examples respectively. The SCAN ADDJUMP and AROUNDRIGHT dataset are released under BSD License.

COGS (Kim and Linzen, 2020) challenges models to parse a diverse set of natural language sentences into their corresponding logical forms based on lambda calculus to accurately reflect the semantic representation of the natural sentence. The train, validation, and test sets have 24155, 4200, and 21000 examples respectively. The COGS dataset is released under MIT License.

CoGnition (Li et al., 2021) is an English-to-Chinese translation dataset with a synthetic OOD test set, where each sentence contains novel compositions of seen structures and constituents. The train, validation, and test sets have 196246, 2160, and 10800 examples respectively. The CoGnition dataset is released for public use and has no explicit license.

WMT. We use WMT17 English↔German (Bojar et al., 2017) and WMT14 English↔French (Bojar et al., 2014) translation tasks. WMT17 En↔De has 3961179 English-German sentence pairs in the training set, 40058 sentence pairs in the validation set, and 3003 sentence pairs in the test set. WMT17 En↔Fr has 35762532 English-German sentence pairs in the training set, 26854 sentence pairs in the validation set, and 3003 sentence pairs in the test set.

B.2 Experimental Setup

Semantic parsing experiments. We use a 3-layer Transformer encoder and a 3-layer Transformer decoder with 4 heads per layer, a hidden size of 256, and a feedforward size of 512. We share input and output embeddings of the decoder. We optimize the model using Adam (Kingma and Ba, 2015), with $\beta_1 = 0.1$, $\beta_2 = 0.98$. All models are trained for 100,000 steps and we choose the best checkpoint on validation set for evaluation. On SCAN tasks where the original vocab size is small (17 for source and 10 for target, including special tokens), we try source codebook sizes [4,6,8] and target codebook sizes [3,4,5], and end up using 6 codes for quantizing source tokens and 4 codes for quantizing target tokens. On COGS with 747 source tokens and 687 target tokens, we try source and target codebook sizes [8,16,32], and end up using 32 codes for quantizing source tokens and 16 codes for quantizing target tokens.

Machine translation experiments. We use a 6-layer Transformer encoder and a 6-layer Transformer decoder with 8 heads per layer, a hidden size of 512, and a feedforward size of 1024. It has the same size as the Transformer used in Yin et al. (2023). We share input and output embeddings of the decoder. The model parameters are optimized by Adam (Kingma and Ba, 2015), with β_1 = 0.1, β_2= 0.98. All models are trained for 1 million steps on CoGnition and 2 million steps on WMT training sets. We then choose the best checkpoint on the validation set for evaluation. During decoding, we use a beam size of 5 and a maximum generation length of $ax + b$ where a=1.2 and b=10. On CoGnition with 2004 source English tokens and 5500 target Chinese tokens (including special tokens), we try source and target codebook sizes [16,32,64,128], and end up using 64 codes for quantizing source tokens and 32 codes for quantizing target tokens. On WMT tasks, we follow the tokenization and preprocessing steps in fairseq[16], and use 16 codes each for quantizing source tokens and target tokens.

C Additional Analyses

C.1 Probing the Code Embeddings

In Sec. 3.1, we argue that word embeddings x are being quantized into codes z largely based on their syntactic function in a linguistic structure. To support this argument, we conduct a probing experiment on the CoGnition dataset by training a linear classifier on top of the codes z (frozen) to predict the Part-of-speech (POS) tag of the corresponding words x. We find that this classifier can correctly predict the POS tag in 40.8% of the cases. Given the fact that z contains no context information and thus is inherently limited in predicting the finer POS tags (whether "like" is a verb or a preposition), the result of this probing experiment supports our argument that the quantized embeddings (code) z encode a large number of syntactic roles of lexical features.

C.2 Visualizing COGS Embedding Space

In Fig. 7, we show the t-SNE of the source embedding matrices learned on COGS. Compared to SCAN, COGS has a much larger vocabulary (748 VS 13) and more diverse syntactic structures. Again, in Fig. 7c we show that SoVQ can cluster the word embeddings based on their syntactic functions. For example, the "red cluster" (in the middle of 2-d t-SNE space) is mostly made of names like "Andrew" and "Olivia". The "yellow cluster" on the right side is comprised of verbs in their past participle forms (e.g., "smiled, rolled"). The "green" cluster at the bottom includes animals like "fish", "bear", and "giraffe". In Table 5, we show 10 words sampled from the source vocabulary and their closest neighbors in the 2-d t-SNE space. One interesting finding is that the word 'can' is clustered with nouns like 'block', 'bee', and 'ring'. This is because in the COGS training set, 'can' is only used as a noun and never used as a modal verb.

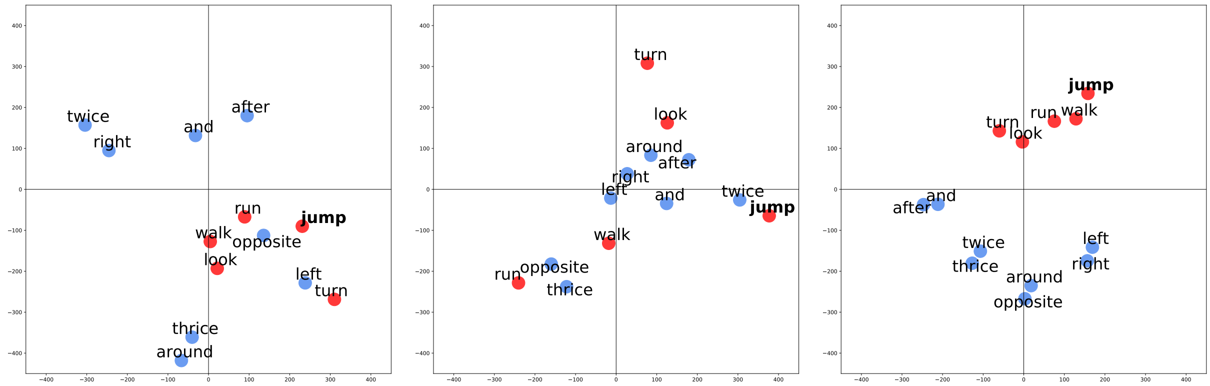
The embedding space learned by the Transformer baseline, on the other hand, does not demonstrate any patterns that can connect the distribution and the syntactic role of a word Fig. 7a. We can observe that sometimes words that are different tenses of the same verb (e.g., "given" and "gave" in the red circle) or have connections in their semantics (e.g., "sleep" and "bed" in the blue circle) are clustered together. The Transformer with vanilla Vector Quantization, although learns a more cluster-separated embedding space Fig. 7b, also does not demonstrate any noticeable similarities of words within a cluster.

Overall, based on the difference in how words are clustered in the embedding space, we can state that Structural-oriented Vector Quantization (SoVQ) can effectively cluster words based on their syntactic functions.

C.3 Visualizing the attention patterns of vanilla Transformer.

As is discussed in Sec. 2, we train a vanilla Transformer model with 3 layers and 4 heads on the SCAN ADDJUMP training set of different complexities (Zhou et al., 2023). For example, the 20x augmented training set has 20 times more primitives (excluding jump) that appear in a variety of syntax structures. In Fig. 8, we visualize its attention maps from processing two separate expres-

[16] https://github.com/facebookresearch/fairseq

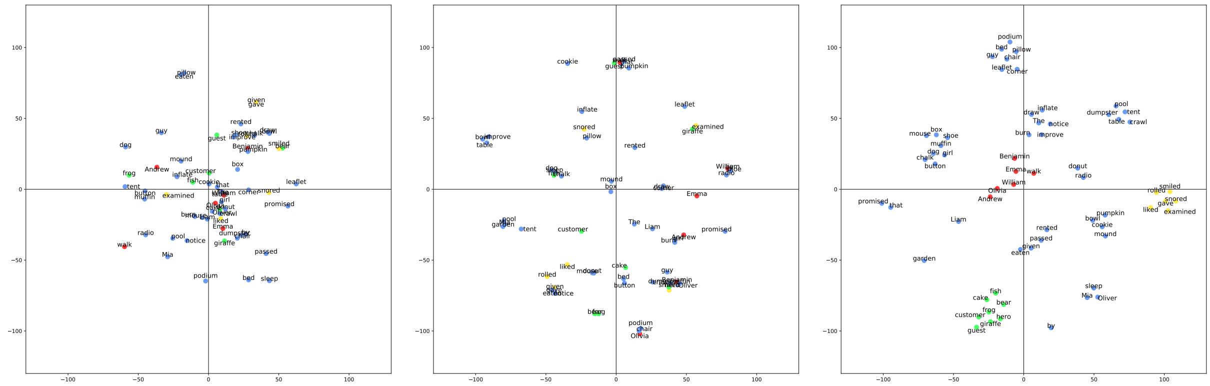


(a) Source embeddings trained with no quantization.

(b) Source embeddings trained with vanilla VQ.

(c) Source embeddings trained with SoVQ.

Figure 6: T-SNE visualization of embeddings learned on SCAN ADDJUMP dataset (Lake and Baroni, 2018), with 6 clusters used in VQ.



(a) Source embeddings of the Baseline.

(b) Source embeddings with vanilla VQ.

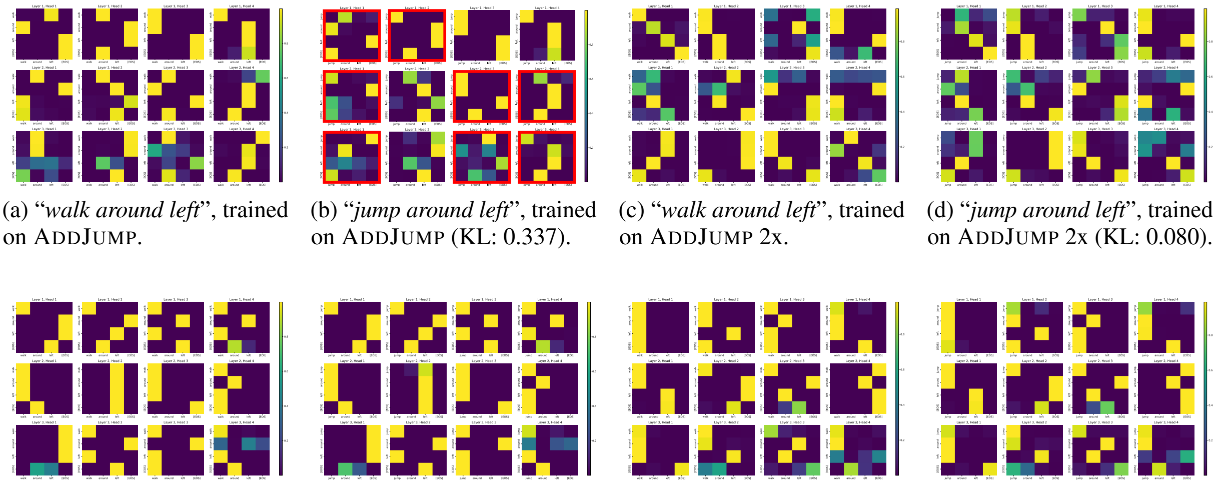
(c) Source embeddings with SoVQ.

Figure 7: T-SNE of embeddings learned on COGS (Kim and Linzen, 2020), with 32 clusters used in VQ.

Source Word	Closest Words
William	Sophia, Riley, Carter, Nora, Madison, John, Jack, Lillian, Sebastian, Christopher
draw	float, love, scream, double, see, shark, help, roll, frown, worship
ate	gave, wanted, noticed, laughed, smirked, hoped, talked, napped, doubled
bear	manager, writer, cow, warrior, governor, crocodile, bicycle, boulder, bag, cloud
cookie	piano, rug, car, banana, melon, bench, bottle, bible, storage, turtle
improve	redden, poke, pierce, discover, throw, toss, slide, freeze, disintegrate, Paula
bed	taxi, pot, sphere, cot, couch, bunker, backpack, glacier, vehicle, bin
preferred	jogged, smiled, sketched, craved, touched, gasped, yearned, supported, saw, crumple
teacher	strawberry, raisin, beast, soap, monster, clock, rose, child, lawyer, chief
can	block, bee, ring, blender, tripod, seat, jacket, dog, donkey, beer
table	stage, speaker, desk, barrel, boat, trunk, house, room, stand

Table 5: Words sampled from the source vocabulary of COGS and their closest words in the 2-d t-SNE space.

sions: "*walk around left*", which is in the training set, and "*jump around left*", which is excluded from the training set. We also show the generalization accuracy and Kullback–Leibler (KL) divergence between the two attention distributions, averaged over all heads and layers. In summary, we show that as the number of distinct primitives increases, the KL divergence between the attention distributions of these two examples decreases. This improvement in attention similarity positively correlates with the improvement of the generalization accuracy. With 20x more lexical primitives than the original training set, the KL divergence reaches 0.001 and the accuracy reaches 100%. This observation provides



(a) "*walk around left*", trained on ADDJUMP.

(b) "*jump around left*", trained on ADDJUMP (KL: 0.337).

(c) "*walk around left*", trained on ADDJUMP 2x.

(d) "*jump around left*", trained on ADDJUMP 2x (KL: 0.080).

(e) "*walk around left*", trained on ADDJUMP 20x.

(f) "*jump around left*", trained on ADDJUMP 20x (KL:0.001).

(g) "*walk around left*", trained on ADDJUMP 200x.

(h) "*jump around left*", trained on ADDJUMP 200x (KL:0.001).

Figure 8: Encoder's attention maps and their average KL-divergence between the two examples, from the vanilla Transformer trained on the SCAN ADDJUMP datasets of different number of distinct primitives (Zhou et al., 2023) (original, 2x, 20x, 200x), where the model achieves 3.67%, 15.78%, 100%, and 100% accuracy on the entire test set respectively.

a mechanistic explanation of how a Transformer trained on more complex data acquires better out-of-domain generalization to novel combinations of structure and lexical entities.

C.4 Analyzing the Attention Pattern of SRL.

We show the example pairs used in Sec. 5.3. We collect a total of 48 pairs of source sentences from the test set. The source sentences within each pair are quantized into the same sequence of clusters (e.g., "*he stopped every girl.*" and "*he found each child.*") by our *SQ-Transformer*.

D More Discussion

D.1 Polysemy

Vector Quantization on token embeddings assigns a token to one of several mutually exclusive classes (Eqn. 1). Therefore, it does not explicitly consider polysemy, which requires assigning a token to potentially multiple syntactic classes based on its context. For example, depending on the context, "sign" could belong to the verb class or the noun class. Empirically we find that words with multiple syntactic roles indeed have non-zero quantization probabilities to multiple classes: after being trained on the WMT En-De, the embedding of "sign" has a 0.3 probability of being quantized to a class mostly including verbs, and a 0.65 probability of being quantized to a class mostly including nouns. This is because, while the cluster inference posterior $q(z|x)$ is conditioned on the lexical embeddings only, the cluster prediction prior $p(z|\hat{z})$ is conditioned on the context and thus can predict the two distinct syntactic roles of the word "sign". As a result of minimizing their cross-entropy, the posterior $q(z|x)$ also learns a bi-modal class distribution for $x =$"sign".

With that being said, since a token can only be assigned to the top class (noun class for "sign") before any contextualization, the verb sense of "sign" is lost after SoVQ. Therefore, to accurately quantize polysemies like "sign", future works can explore quantizing contextualized representation, like the outputs after the first 3 layers, of words. This can potentially lead to better-quality, context-aware code representation used in the downstream SAL/SRL.

D.2 Scaling Up Model and Data

Scaling up the training data. In this work, we trained *SQ-Transformer* on datasets up to the scale of WMT machine translation datasets, which are the most popular benchmarks in evaluating seq2seq neural architectures (as used in Vaswani et al. (2017)). We also support our main arguments via proof-of-concept experiments based on semantic parsing. We do not train *SQ-Transformer* on larger web-scale, language-modeling datasets and leave

Sentence 1	Sentence 2
She chose another child he liked .	She chose another clown he liked .
She also found every small girl .	She also met each small clown .
He took every large clown .	He caught every special clown !
He heard each large girl .	He saw each small girl .
He heard every silly girl .	He found every dirty girl .
The man took every large clown .	The cat caught every special clown .
He stopped every girl .	He met each girl .
She saw the silly clown .	She watched the dirty girl .
She met each small clown .	She watched every large girl .
She invited the large clown .	She saw the special child .
The smart doctor he liked was very proud .	The dirty dog he liked was extremely excited .
He invited each small boyfriend on the floor .	He found every large boyfriend on the floor .
He left every small girl he liked .	He caught each large child he liked .
Taylor met the special girl .	Taylor heard the small clown .
He visited the large child he liked .	He heard the large girl he liked .
I chose another clown he liked .	I chose another child he liked .
She visited each silly boyfriend on the floor .	She found every dirty boyfriend on the floor .
He hated the large clown on the floor .	He applied to the large doctor on the floor .
Taylor hated the small building .	Taylor hated the large building .
Any red doctor at the store jumped back .	Any small farmer at the store pushed him .
She woke any large building on the floor .	She woke any small building on the floor .
I chose each large boyfriend on the floor .	I chose each small car on the floor .
His neighbors heard the special clown on the floor .	His friend heard the small clown on the floor .
He looked under any small chair on the floor .	He looked under any small chair on the floor .
There was a hurricane headed towards each small doctor .	There was a hurricane headed towards every small doctor .
Taylor hated every silly bee .	Taylor went to each empty bee .
Taylor was sad about every small building .	Taylor was excited about every small building .
She had everything taken care of except every small chair .	She had everything taken care of except each large apartment .
Taylor visited each silly boyfriend on the floor .	Taylor found every silly boyfriend on the floor .
Taylor lost another small girl on the floor .	Taylor lost another small clown on the floor .
Taylor watched every dirty girl he liked .	Taylor saw every silly clown he liked .
Taylor chose each small girl he liked .	Taylor caught each large child he liked .
Taylor woke each large clown for school .	Taylor woke every small clown for school .
Taylor visited the small child he liked .	Taylor heard the large girl he liked .
Taylor caught any large car on the floor .	Taylor took any small boyfriend on the floor .
Taylor found every large child on the floor .	Taylor heard each small clown on the floor .
Taylor watched each dirty clown on the floor .	Taylor heard every dirty clown on the floor .
He smiled and gave each car a free popcorn .	He smiled and gave every boyfriend a free popcorn .
She invited all the girls except any small building on the floor .	She invited all the girls except any small farm on the floor .
Taylor hated every silly bee he liked .	Taylor hated each dirty bee he liked .
Taylor hated any large bee on the floor .	Taylor hated any small bee on the floor .
I woke the silly child on the floor up to give him a sandwich .	I woke the dirty clown on the floor up to give him a sandwich .
Another smart doctor he liked got so bad that she could n't stand it .	Another smart doctor he liked got so bad that she could n't stand it .
Except each empty apartment he liked , she took it out to show to a friend .	Except every empty airplane he liked , she took it out to show to a friend .
When i got home , i did all my homework except each empty apartment he liked .	When i got home , i did all my homework except every empty airplane he liked .
When i got home , i did all my homework except the quiet farm he liked .	When i got home , i did all my homework except the empty building he liked .
Taylor stayed inside the small building on the floor , even though a storm was coming .	Taylor stayed inside the large farm on the floor , even though a storm was coming .
As soon as i was about to take a bath , i saw a light inside any empty car he liked .	As soon as i was about to take a bath , i saw a light inside any quiet car he liked .

Table 6: Source sentences collected from the CoGnition (Li et al., 2021) compositional generalization test set. The sentences within each pair are quantized into the same sequence of clusters by our *SQ-Transformer*.

this to future work. We hope the promising results of *SQ-Transformer* can inspire more researchers to experiment with it on web-scale data.

Scaling up *SQ-Transformer*. In this work, we investigate *SQ-Transformer* up to 6 layers, 8 heads, and 512 hidden dim, same as the model size in Vaswani et al. (2017). To further scale the model up, there are two main challenges. First, we use two separate Transformer decoders as the cluster prediction prior $p(z|\hat{z})$ (Eqn. 4), which brings memory overhead. However, the cluster predictors can be much smaller than the main Transformer network because the structure-oriented latent code space (16 for WMT En-De) is much smaller than the main network's semantic-enriched word space (40356 for WMT En-De). It is also possible to share the parameters between the main network and the cluster predictors. Second, most LLMs use some types of byte pair encoding (BPE) tokenization to build a universal, sub-word level vocabulary. As a result, the structural role of a sub-word (e.g., '*token-*') could depend on its suffix: "*token-ize*" is a verb while "*token-ization*" is a noun. This is similar to the polysemy problem at the word level (discussed in Appendix D.1), but is much more frequent at the sub-word level. So far, our proposed Structure-oriented Vector Quantization (SoVQ) is only performed on the sub-word embeddings. Therefore, without any context information, SoVQ cannot accurately quantize '*token*' to either the verb class or the noun class. Nonetheless, we still showed SoVQ's effectiveness in WMT machine translation datasets (tokenized with BPE). This suggests that the SoVQ is robust to some ambiguous sub-words like the prefix '*token*'. However, further scaling SoVQ to web-scale data with a much larger sub-word vocabulary could potentially be challenging, since some of the special sub-words are not meaningful by themselves.[17] To address this challenge, we believe future works can experiment with quantizing the contextualized representation (e.g., the output representations of the first k layers) rather than quantizing the sub-word embeddings.

D.3 How does Transformer Generalize Compositionally?

It has long been argued that neural networks are associative devices that cannot capture systematic compositionality (Fodor and Pylyshyn, 1988; Marcus, 1998; Fodor and Lepore, 2002; Marcus, 2003; Calvo and Symons, 2014). Specifically, Fodor and Pylyshyn (1988) claimed that "*in traditional Associationism, the probability that one Idea will elicit another is sensitive to the strength of the association between them. ... Associative strength was not, however, presumed to be sensitive to features of the content or the structure of representations per se. Similarly, in Connectionist models, the selection of an output corresponding to a given input is a function of properties of the paths that connect them.*" As a result, they further stated that ***"The syntactic/semantic structure of the representation of an input is not presumed to be a factor in determining the selection of a corresponding output since, as we have seen, syntactic/semantic structure is not defined for the sorts of representations that Connectionist models acknowledge."*** After we showcase the systematic behavior of *SQ-Transformer*, readers might ask "how does our method overcome the inherent limitation of Connectionist models elicited in Fodor and Pylyshyn (1988)?"

First, the statement above made an important assumption about neural networks (i.e., Connectionist models) that syntactic/semantic structure does not determine the strength of association between neurons (through the form of attention or full connection). We agree that the strength of the association is decided by the correlations a neural model observed in data. Thus, models like Transformers and RNNs fail to execute "*jump twice*" because where "*jump*" and "*twice*" are never seen together in training and models are insensitive to the syntactic structure.

However, we argue that with the proper regularization (e.g., SoVQ and SRL) to the intermediate representations (incl. embeddings and layer outputs), neural models can take the sentence structure into account when determining the strength of inter-neuron association via attention weights or MLP connection. **This is because neural models are not inherently limited to capturing shallow, word-level correlations.** As is shown in Hewitt and Manning (2019), the attention maps can also be sensitive to the common, although latent, dependency between words, which is simply a kind of statistical correlation between multiple latent and explicit factors (word, position, order, etc). Therefore, the resulting representations also encode rich structural information. The extent to which the model can be sensitive to the structure depends on the data complexity (Zhou et al., 2023), model architecture (Murty et al., 2023b), and regularization (Jiang and Bansal, 2021; Yin et al., 2023). For example, there is a statistical correlation between the input word "*twice*" and the latent output structure (always repeating the action preceding "*twice*" 2 times). We showed that *SQ-Transformer* can very well capture this association systematically in its

[17] For example, "<0xF0><0x9F><0xA6><0x99>" forms a "llama" emoji but each sub-word itself does not have a fixed role in linguistics.

attention maps.