A BACKGROUND ON GROUP THEORY

As a reminder, we will present a series of basic definitions from group theory. These definitions can serve as points of reference from the main body of the paper.

Definition A.1. A *group* G is a set equipped with a binary operation $\mu : G \times G \to G$, usually denoted just by $(g_1, g_2) \mapsto g_1 g_2$, satisfying the following axioms:

- **Associativity**: For all $g_1, g_2, g_3 \in G$, $(g_1 g_2)g_3 = g_1(g_2 g_3)$.
- **Identity Element**: There exists an element $1 \in G$, called the *identity element*, such that for all $g \in G$, $g1 = 1g = g$.
- **Inverse Element**: For every element $g \in G$, there exists an element $g^{-1} \in G$, called the *inverse element* of g, such that $gg^{-1} = g^{-1}g = 1$.

An arbitrary group can be expressed as a quotient (Definition A.6) of a free group.

Definition A.2. Let X be a set. The *free group* F on X is a group defined as follows: F consists of all reduced words (finite sequences) on elements of S and their formal inverses, equipped with the operation of concatenation followed by reduction. The reduction operation eliminates pairs of the form xx^{-1} and $x^{-1}x$. The identity element of F is the empty word.

Free groups, in comparison with arbitrary groups, have a simple algebraic structure, so as their *subgroups*, i.e. subsets closed under group operations.

Theorem A.3 (Nielsen–Schreier theorem, (Kargapolov & Merzljakov, 1979, Section 14.3))**.** *Every subgroup of a free group is free.*

Subgroups R_i of Definition 1, that we are dealing with, are defined in a way that they are closed under conjugation operation.

Definition A.4. Let G be a group. A subgroup N of G is called a *normal subgroup* if and only if for every $g \in G$ and $n \in N$, the element $n^g = g^{-1}ng$ is also in N.

Definition A.5. Let G be a group and S be a subset of G. The *normal closure* of S in G, denoted by $\langle S \rangle^G$, is the smallest normal subgroup of G that contains S. It is the intersection of all normal subgroups of G containing S.

Wu's formula (4) involves a quotient of an intersection of R_i's by their symmetric commutator subgroup.

Definition A.6. Let G be a group and N be a normal subgroup of G. The *quotient group* of G by N, denoted by G/N, is the set of *cosets* of N in G, i.e. subsets of G of the form $gN = \{gn | n \in N\}$, with group operations induced from G.

Finally, some of our methods rely on the presentations of R_i as kernels of *homomorphisms* d_i, i.e. maps between groups that preserve group operations, see also Formula (8).

Definition A.7. A *kernel* of a group homomorphism $f : G \to H$ is the following subset:

$$\ker f = \{g \in G | f(g) = 1\}.$$

Note that the kernel of a group homomorphism $G \to H$ is always a normal subgroup of G.

Throughout the paper, we utilize the notion of commutators and conjugations, which were introduced in Section 3, to provide a shorter representation of words in a free group. There is a number of *identities*, involving commutatrors, that hold for all elements of any group, for instance:

$$[x, zy] = [x, y][x, z][x, z, y]$$
$$[xz, y] = [x, y][x, y, z][z, y]$$
$$[x, y^{-1}, z]^y [y, z^{-1}, x]^z [z, x^{-1}, y]^x = 1.$$

The last identity, called the *Hall-Witt identity* is a "non-abelian" version of the Jacobi identity in Lie algebras. As mentioned at the end of Section 4.1, these identities enable us to express every

commutator as a product of iterated commutators **of the same length**. For example, for commutator $[[a,b],[c,d]]$, the Hall-Witt identity reads:

$$[a,b,[c,d]]^{b^{-1}}[b^{-1},[d,c],a]^{[c,d]}[c,d,a^{-1},b^{-1}]^{a}=1$$

and since

$$[b^{-1},[d,c],a]=[d,c,b^{-1},a^{[b^{-1},[d,c]]}]^{-1},$$

we get

$$[[a,b],[c,d]]=[c,d,a^{-1},b^{-1}]^{-ab}[d,c,b^{-1},a^{[b^{-1},[d,c]]}]^{[c,d]b}.$$

B BACKGROUND ON SIMPLICIAL HOMOTOPY THEORY

In this section we will provide a brief overview of the historical development of the concept of homotopy groups (see also (Hilton, 1988)). Subsequently, we will delve into the basics of simplicial homotopy theory and the description of Wu's formula within the context of simplicial groups.

B.1 HISTORY OVERVIEW

As a distinguished area of mathematics, algebraic topology was surfaced since the end of XIX century. During this time, mathematicians realized that certain important properties of geometric objects, like properties of vector fields on a manifolds or solutions of differential equations on them are controlled by *topological* properties of the underlying objects of study, rather than geometric or analytic characteristics. This realization prompted the need for algebraic invariants that could discern between classes of spaces with similar topological properties and aid in their classification. Historically, one of the first invariant was *homology* H_* of space, shortly followed by a fundamental group π_1.

Such invariants were able to distinguish spaces *up to homotopy equivalence*. Specifically, two maps $f,g:X\to Y$ are called homotopic, if there is a family of maps $h_t:X\to Y,\ t\in[0,1]$, continuous in t, which "connects" f to g in a sense that $h_0=f,\ h_1=g$. Similarly, two spaces X,Y are *homotopy equivalent* if there exist maps $X\leftrightarrows Y$, which compositions with each other are homotopic to identities.

Generalizing π_1, the series of higher invariants π_*, called the *homotopy groups*, were defined as homotopy classes of maps from higher dimensional spheres S^n to a space of interest. Stronger than homology, homotopy groups capture the substantial part of a homotopy type of space, which we nowadays call a *weak homotopy type*. Extremely difficult to compute and comprehend, till today, homotopy groups remain the main object of interest in algebraic topology. Various flavors of homotopy theory have been developed, including stable homotopy, v_n-periodic homotopy and others, each with corresponding homotopy groups that are more tractable and accessible than the classical ones.

In parallel with the development of various algebraic invariants for classifying spaces, the concept of space itself has undergone several formalizations within homotopy theory. Initially, the notion of a space referred to a *topological space*, which consisted of a set equipped with a chosen collection of subsets called a *topology*. Later the concept of an (abstract) simplicial complex emerged as a combinatorial notion of space, suitable for defining homology groups. This notion was subsequently refined to that of a simplicial set.

Homotopy theory of simplicial sets, which is both combinatorial and algebraic in nature, now serves as the foundation for more advanced concepts in abstract homotopy theory, such as simplicial model categories, quasicategories, simplicial localizations, and others. Within the scope of the current paper, a specific type of simplicial sets called *simplicial groups* holds particular significance. Simplicial groups serve as a crucial link between simplicial homotopy theory and group theory, offering a concrete, algebraic, and computational-friendly framework.

B.2 SIMPLICIAL GROUPS AND WU'S FORMULA

We will now describe the relevant part of simplicial homotopy theory in greater detail. Simplicial set (formally described as a functor $\Delta^{op}\to\mathrm{Set}$) can be visualized (Mac Lane, 2013)[Ch. 7.5] as a

diagram of sets

$$X: \qquad \pi_0 X \xleftarrow{d_0} X_0 \underset{\xleftarrow{d_1}}{\overset{\xleftarrow{d_0}}{\xrightarrow{s_0}}} X_1 \underset{\xrightarrow{s_1}}{\overset{\xleftarrow{d_0}}{\xrightarrow{s_0}\ \xleftarrow{d_1}}} X_2 \qquad \dots,$$

with maps d_i, called *faces* and s_j, called *degeneracies* which satisfy the *simplicial identities*:

$$\begin{aligned} d_i d_j &= d_{j-1} d_i,\ i < j \\ s_i s_j &= s_j s_{i-1},\ i > j \\ d_i s_j &= \begin{cases} s_{j-1} d_i,\ i < j \\ \mathrm{id},\ i \in \{j, j+1\} \\ s_j d_{i-1},\ i > j+1 \end{cases} . \end{aligned} \tag{6}$$

Drawing intuition from simplicial complexes, each X_n can be thought of as a set of n-simplices, with maps $d_i : X_n \to X_{n-1}$ indicate an i-th face of an n-simplex and maps $s_j : X_{n-1} \to X_n$ indicate ways how to consider $(n-1)$-simplex as a (degenerate) n-simplex.

Simplicial sets were introduced by Eilenberg & Zilber (1950) (originally under the name of semi-simplicial complexes) for the needs of formalization of homology theory. When the abstract homotopy theory became more articulated in the fundamental works of Gabriel & Zisman (2012) and Quillen (2006), it was shown that the corresponding homotopy theory is equivalent to the classical homotopy theory of topological spaces. Although for some concrete work in homotopy theory simplicial sets may better suited than topological spaces/cell complexes, regarding computations of homotopy groups, bare simplicial sets have some difficulties when it comes to computation of homotopy groups. For instance, the naive notion of homotopy is not an equivalence relation for all simplicial sets.

One of the possible solutions is to consider *simplicial groups*, i.e. simplicial sets which have a group structure on the sets of their simplices, such that the maps d_i, s_j preserve this structure. A notable advantage of simplicial groups is the very explicit description of their homotopy groups: if G is a simplicial group, then

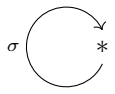
$$\pi_n G = \frac{\cap_{i=0}^{n} \ker d_i}{\operatorname{im} \partial_{n+1}}, \tag{7}$$

where ∂_{n+1} is a restriction of d_{n+1} to the intersection of kernels of the next dimension. So the homotopy groups of G are identified with the quotients of intersections of kernels of face maps d_i by the images of the last face maps (restricted to a suitable intersection).

To get the Wu's formula for $\pi_n S^2$ from (7), we will consider a particular simlicial group (more precisely, the functor $s\mathrm{Set} \to s\mathrm{Gr}$) called the *Milnor construction* (Curtis, 1971)[Ch. 4]. Let X be a connected simplicial set with a basepoint $*$. The Milnor construction for X, denoted by $F[X]$ is a simplicial group obtained by level-by-level-wise application of the free group functor to X, quotient by the relation $* = 1$. This means that n-simplices in Milnor's construction $F[X]$ form free group on n-simplices X_n with the identity of the group identified with a basepoint $*$. The reason for introducing this construction is that for a connected simplicial set X the Milnor's construction $F[X]$ has a homotopy type of loops over the suspension of X. For $X = S^1$ on the level of homotopy groups it means that

$$\pi_n F[S^1] = \pi_{n+1} S^2.$$

The simplicial circle S^1 consists of one non-degenerate 0-simplex, which is a basepoint $*$ and one non-degenerate 1-simplex σ (plus the degeneracy $s_0(*)$), and there are no non-degenerate simplices in all higher dimensions.



Now simplicial identities (6) show that

$$F[S^1]_n = F[\{x_0, \dots, x_{n-1}\}]$$

and faces are given by

$$d_0(x_j) = \begin{cases} 1,\ j = 0 \\ x_j,\ j > 0 \end{cases} \qquad d_i(x_j) = \begin{cases} x_j,\ j < i \\ x_{i-1}x_i,\ j \in \{i-1, i\} \\ x_{j-1},\ j > i \end{cases} \qquad d_n(x_j) = \begin{cases} x_j,\ j < n-1 \\ 1,\ j = n-1 \end{cases} .$$

It is shown in (Wu, 2001) that the kernels $\ker d_i$, after an appropriate change of basis and relabeling, can be identified with normal subgroups

$$\ker d_0 = R_0 = \langle x_1 \dots x_n \rangle^F \ker d_i = R_i = \langle x_i \rangle^F, i > 0, \tag{8}$$

and the image in the denominator of Formula (7) with their *symmetric commutator subgroup*

$$\operatorname{im} \partial_{n+1} = [R_0, \dots, R_n]_S = \prod_{\sigma \in \Sigma_{n+1}} [R_{\sigma(0)}, \dots, R_{\sigma(n)}].$$

This is the idea behind the Wu's formula.

C Implementation details

To operate with words in free groups we implemented a small `Python` module `freegroup`. In this module we use lists of integers as the representation of words with the following mapping: $x_k \mapsto k, x_k^{-1} \mapsto -k$.

As a base for our models we used GPT2 (Radford et al., 2019) implemented by `huggingface` (Wolf et al., 2020). For $n = 3$ we used the model with $\sim 2 \cdot 10^6$ parameters and for $n = 4, 5$ we used $\sim 50 \cdot 10^6$ parameters. We employed standard training framework with iterating through available data and gathering examples to batches. We trained models for $2 \cdot 10^5, 5 \cdot 10^5$ iterations with batches of sizes $64, 16$ for $n = 3$ and $n = 4, 5$ cases, respectively. For the optimization we used AdamW (Loshchilov & Hutter, 2019) implemented by `PyTorch` (Paszke et al., 2017) with learning rates: 10^{-4} for $n = 3$ and 10^{-6} for $n = 4, 5$. For the inference we used generation algorithms implemented by `huggingface`. During sampling `top_p` $= 0.9$; during beam search `num_beams` $= 5$, `repetition_penalty` $= 1.2$.

Dataset parameters are the following:

- maximal length of a word = $200, 400, 600$,
- maximal lengths of a word from R_0 are $10, 9, 8$ for $n = 3, 4, 5$ respectively,
- maximal lengths of a word from $R_i,\ i > 0$ are $30, 30, 30$ for $n = 3, 4, 5$ respectively.

D Additional experiments, models and examples

D.1 Other models

In addition to the deep learning methods discussed in the main body of the paper, we explored several other trainable and non-trainable machine learning approaches for word generation.

Fixed embeddings Alongside the development of deep learning models, which are in fact suitable deep feature extractors coupled with a classifiers, we also investigated the usefulness of non-trainable embeddings and experimented with different optimizers and machine learning classifiers applied on top of them. The underlying motivation for this investigation was to transform the problem setting from a discrete one to a continuous one, as most optimization and machine learning methods operate more effectively in continuous domains.

The variant of the non-trainable embeddings that we investigated is based on the matrix representation of free group. Using the group homomorphism $\rho : F_2 \to SL_2(\mathbb{R})$, called *Sanov representation* (Zubkov, 1998)

$$x \mapsto \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix},\ y \mapsto \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \tag{9}$$

the free group F_2 on two generators x and y can be embedded in a space of two-by-two matrices with real coefficients (in fact, into a subgroup of matrices M with $\det M = 1$)

The embedding of a free group F on an arbitary number of generators n is obtained by inclusion

$$F \to F_2,\ x_i \mapsto [x, y^i].$$

Remark D.1. Note that for a word $w \in F$ the magnitudes of coefficients in the matrix $\rho(w)$ depends exponentially on the length of w, making this embedding practically unfeasible. Nevertheless we tested an optimization on this embedding space to evaluate a potential of non-trainable embeddings on a small scale.

One of the advantages of matrix embeddings is that the homomorphisms $d_i : F \to F$, such that $\ker d_i = R_i$, as in Formula (8) can be reinterpreted as a matrix operation of nullifying i-th column denoted f_i, $i > 0$. At the same time, since ρ preserves the group operation, verifying whether $w \in R_i$ is equivalent to checking if $f_i\rho(w)$ is equal to the identity matrix I.

We investigated various optimization procedures on this feature space, however, regrettably, the method proved ineffective in generating any words for $n \geq 3$, even when considering the partial intersection $\cap_{i=1}^{n} R_i$.

Remark D.2. We have also considered posibility of hyperbolic embedding, i.e. embeddings of the (compact region of) Cayley graph of F into hyperbolic plane $\mathbb{H}$. We considered embedding by Sarkar algorithm (Sarkar, 2012) or as an orbit of i under $\rho : F \hookrightarrow SL_2(\mathbb{R})$ acting on the upper half-plane $\mathbb{C}_{\mathrm{Im}\geq 0}$ by Mobius transformations.

Both embeddings, although not isometric, retain certain metric properties of F, where the metric structure of F is determined by the *word metric*. However, during our preliminary investigations, we discovered that the metric structure of F might not be the most suitable setting for optimization in our specific problem.

Activation maximization While experimenting with neural based classifiers for elements in R_i, we investigated the feasibility of methods from the area of interpretable neural networks. Technique that we were studying, described in (Nguyen et al., 2019), involves maximizing the activation of specific neurons in a neural network to generate images that are most likely to activate those neurons. We tried to adopt this method to the free group's word generation as it was done for protein sequences in (Linder & Seelig, 2021).

Following (Killoran et al., 2017), (Linder & Seelig, 2021), we implemented the following approach. If the classifier input is given by sequence of length M over the alphabet of size N, then the latent variable z of size $N \times M$ is representing the probabilities of each token on each position. Suppose that classifier is trained up to a sufficiently high accuracy, then sampling with the probabilities from z and feeding back the resulting words to the classifier, the relaxation procedure can be carried out in the latent space of z.

The appeal of this approach comes from the fact that it can be extended to an ensemble of classifiers. By maximizing the activation of all classifiers within the ensemble simultaneously using the same latent variable z, the approach can be used to generate words from the intersection $\cap_i R_i$ that maximize the activation of output neurons of all classifiers in the ensemble.

The appeal of this approach lies in its potential for extension to an ensemble of classifiers. By maximizing the activation of all classifiers within the ensemble simultaneously using the same latent variable z, the approach can be effectively applied across the entire ensemble.

In the end, the activation maximization method turned out to be incapable of generating at even short non-trivial words from a single R_i. Despite the continuous effort, we were unable to generate words from the intersection using the joint activation maximization of multiple generators.

SeqGAN Another method that has been explored is generative adversarial networks (GANs) (Jabbar et al., 2020), specifically the SeqGAN (Yu et al., 2017) approach. GANs are a type of deep neural network that can generate realistic synthetic data, such as images or text. SeqGAN focuses on generating realistic sequences of data, such as sentences or musical compositions. However, while GANs have shown promising results in some applications (Jabbar et al., 2020) and can be used in similar circumstances (Yao et al., 2019), they did not yield promising results in our problem.

LSTM Ensemble Before developing the general masking approach, based on multi-labels, as mentioned in Remark 4.2, we investigated its particular realization based on the ensemble of deep models. Our initial experiments were based on an ensemble of LSTM cells (Hochreiter & Schmidhuber, 1997), in which each generator G_j is trained on the subgroup R_j or some symmetric commutator subgroup.

In generating the word from the intersection, to sample the next token from the ensemble we used a *temperature sampling*, i.e. the probability of the next token y_k from a sequence y have a distribution

$$y_k \sim \text{Softmax}\left(\frac{1}{\tau}\sum_{i=1}^{N} w_i f_{\theta_i}(y)\right),$$

with weights w_i for each generator f_{θ_i}.

This approach allowed us to get first non-trivial elements from the full intersection. However, it was subsequently surpassed by a more general masking approach, which has also been demonstrated to be more scalable.

D.2 Properties of sampling from R_i

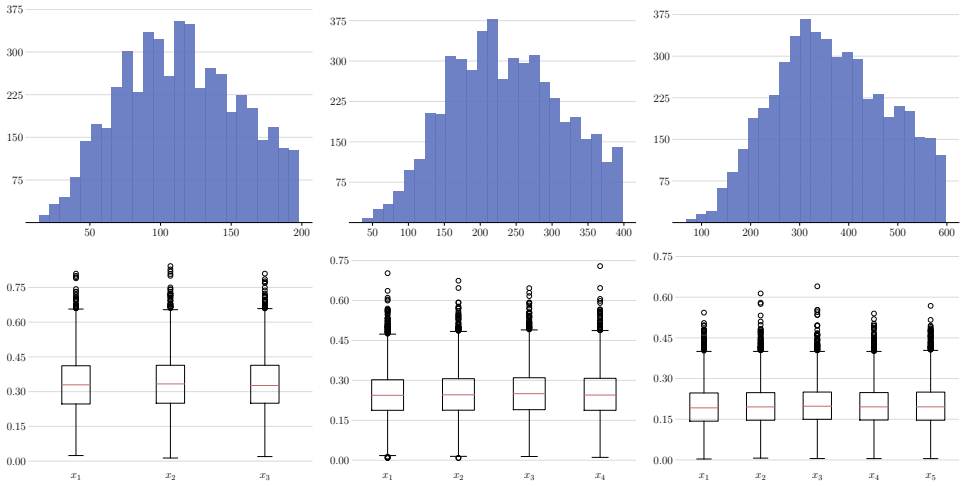


Figure 5: Length distribution (top) and occurrence ratio of generators x_i (bottom) in the training dataset subsample for the $n = 3, 4, 5$.

D.3 Translator model

For a given word $w \in [F, F]$ there are multiple way to present it in form of a product of commutators. Although there are algorithms to find such presentations (Hall, 1950), (Bartholdi et al., 2015), the presentations of long words in terms of basic commutators or as a product of ordinary (not iterated) commutators are bulk and practically unusable. We are looking for short and concise presentations, such as given in Table 1.

To tackle this challenge, we employed an encoder-decoder model (Vaswani et al., 2017) as a translator from a reduced word presentation to the commutator presentation.

The following procedure was implemented to generate the training and validation datasets:

1. A collection of random binary trees with a random number of leaves is generated.
2. The leaves of these trees are replaced with random words from a free group, ensuring that the word length remained within predefined bounds.
3. These modified trees (represented as strings) served as the labels for our model , representing a random commutators that we aim to translate.

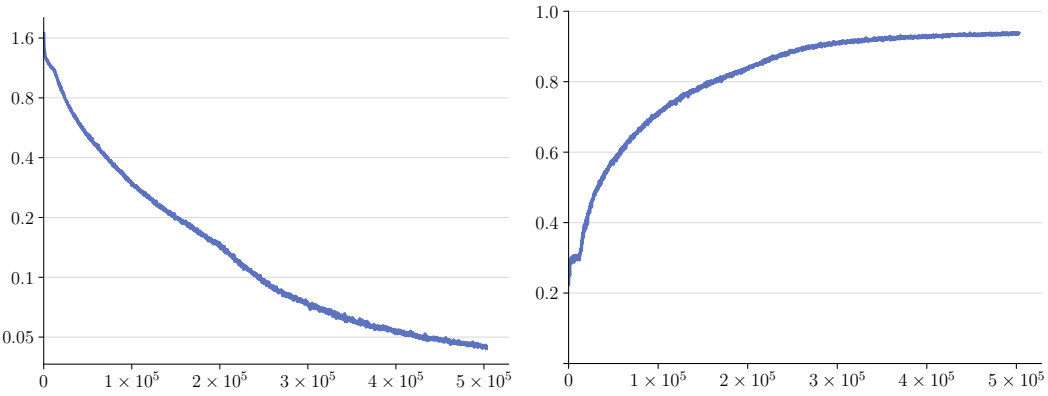


Figure 6: Illustration of commutator translator learning process. Loss with logarithmic scale (left) and BLeU score on validation (right).

4. Inputs, corresponding to the labels are given by reduced words, obtained from labels by opening all commutator brackets.

Subsequently, we employed a standard training method for the translation model. We used BERT (Devlin et al., 2019) with $2 \cdot 10^6$ parameters as an encoder as well as a decoder. We used AdamW with the learning rate $8 \cdot 10^{-5}$ and the batch size equal to 64.

As with the main models (see Section 5), the training and validation datasets are generated in online mode with the following parameters:

- number of generators $n = 4$,
- maximal length of a word = 250,
- maximal length of a leave = 3,
- maximal depth of a tree = 7.

The training process is presented on Figure 6

D.4 Examples of generated words

We provide some examples of generated words for $n = 3, 4$ in commutator presentation for various deep models and inference methods, see Section 4.3 for additional details. The presentation is obtained using translator model described above.

- $n = 3$

$$
\begin{aligned}
&[[y^2, [y^{-1}x^{-1}z^{-1}, [z^2, x]]], x] && \text{(prompt + beam)}\\
&\left.\begin{aligned}
&[[x, y], [yz, x]]\\
&[[xy^{-2}x^{-1}z, y], [z, xy]]\\
&[[[z, xy], y^{-1}xyx], y]
\end{aligned}\right\} && \text{(ignore + sample)}\\
&[[x, z], [x^{-1}, y][y, z]] && \text{(mask + sample)}
\end{aligned}
$$

Note that some of the generated words are not in the symmetric commutator subgroup.

- $n = 4$

$$\left.\begin{array}{r}[[xyzp, [y^4, [x, z]]], p] \\ [[(xyzp)^3, [[p^{-1}, z^2], y]], x]\end{array}\right\} \quad \text{(ignore + beam)}$$

$$\left.\begin{array}{r}[[yzpx^{-3}, x^{-1}], [z^2, [p^2ypy^{-1}, y^4]]] \\ [[[y, p^{-1}zp], xzp^2z^{-1}x^{-1}p], [x^3, zyzpxpxyx^{-1}]]\end{array}\right\} \quad \text{(ignore + sample)}$$

$$\left.\begin{array}{r}[[[[z^{-1}y^{-1}x^{-1}, p], y], x], z^3] \\ [[z^3, [(xyzp)^2, [y^2, x]]], p^3]\end{array}\right\} \quad \text{(mask + beam)}$$

$$[[p^2x^{-1}pxpx^{-1}px, [pzp^{-1}yzy^{-1}, [y, x]]], xyzp] \quad \text{(mask + sample)}$$

$$\left.\begin{array}{r}[[[[p^{-1}z^{-1}y^{-1}x^{-1}, z], y], x], p] \\ [[x^3, [[x^{-1}, p^{-1}z^{-1}y^{-1}], [p, y]]], z^2]\end{array}\right\} \quad \text{(prompt + beam)}$$

$$\left.\begin{array}{r}[[[p^{-1}y^2p, z], p^{-1}xz^{-1}xzp], [xyzp, zp^3xpx^{-1}z^{-1}]] \\ [[[zpx, y], x^{-1}zx^2zx^{-1}z^{-1}], [[x^3yx, y^{-1}], p^3]]\end{array}\right\} \quad \text{(prompt + sample)}$$

D.5 Details on greedy algorithm

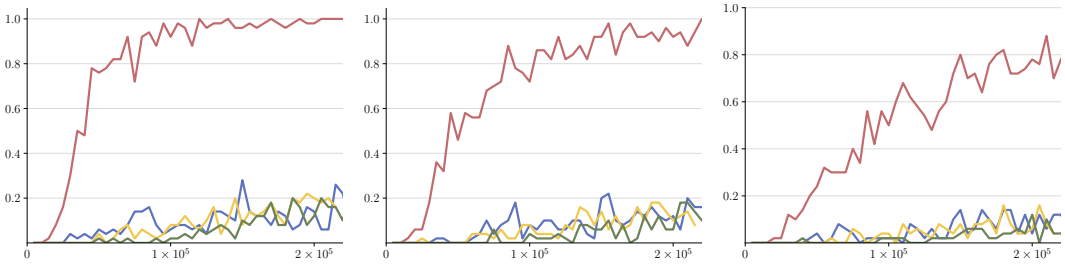
Suppose we have a given prefix $p = xyz$ and we aim to generate a word from the intersection of the normal closures $R_1 = \langle x \rangle^F$, $R_2 = \langle y \rangle^F$, and $R_3 = \langle z \rangle^F$. The corresponding stacks S_i created for each normal closure R_i are $S_1 = [y, z]$, $S_2 = [x, z]$, $S_3 = [x, y]$. In a list of candidates for the next token, the token z^{-1} is excluded because it would reduce the length of the prefix p.

Next, the algorithm assigns points to each token based on the two criteria mentioned in Section 4.2. The token y^{-1} would receive a point because it reduces the length of the S_3. The tokens x and x^{-1} would receive points because they do not increase the length of S_1. Similarly, y, y^{-1}, and z would receive points because they do not increase the length of S_2 and S_3 respectively. The algorithm then selects the token with the highest point value as the next token to add to the prefix. In this case, y^{-1} has the highest number of points, so it would be selected as the next token to add to the prefix.

In the case of a "many-token" normal closure, such as $R_0 = \langle xyz \rangle^F$, tokens receive points based on their ability to bring the top of the corresponding stack S_0 closer to a rotation of the normal closure. In particular, if token x_k is the top of S_0, then the token $x_{(k+1) \bmod (n+1)}$ will receive a point.

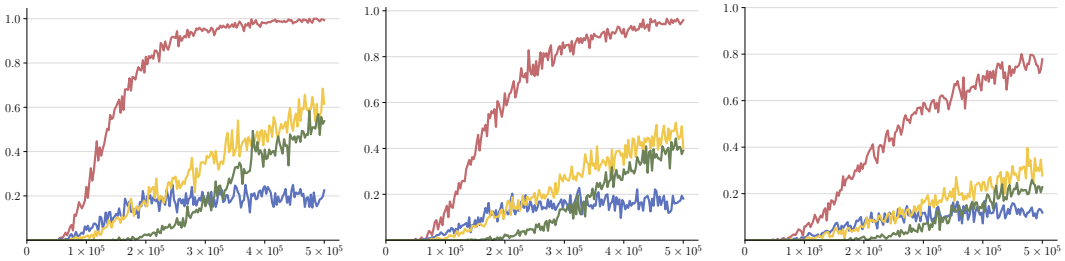
D.6 AUXILARY LEARNING CURVES

In Figures 7-9 we present additional graphs of learning process for various hyperparameters like inference method and prefix length. Note that variable prefix allows more variability in models output as well. Color coding is the same as in Figure 3: red for *negative baseline*, yellow for *prompt*, green for *masking*, blue for *ignore*.



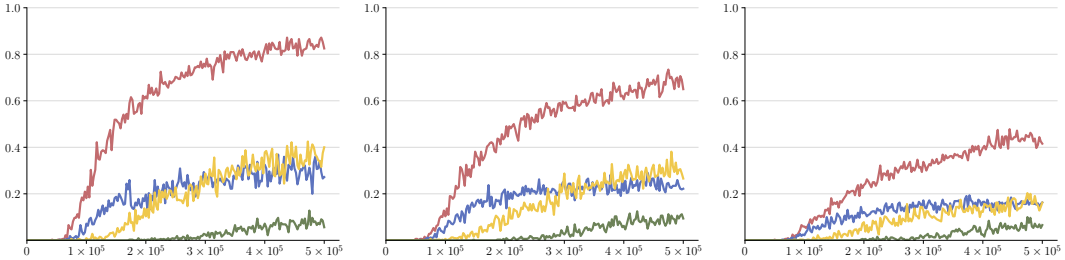
(a) $k = 5$ (b) $k = 7$ (c) $k = 10$

Figure 7: Completion ratio with nucleus sampling for $n = 3$ and various length of prefixes k.



(a) $k = 5$ (b) $k = 7$ (c) $k = 10$

Figure 8: Completion ratio with nucleus sampling for $n = 4$ and various length of prefixes k.



(a) $k = 5$ (b) $k = 7$ (c) $k = 10$

Figure 9: Completion ratio with beam search for $n = 4$ and various length of prefixes k.