
UNIVERSITY OF CAPE TOWN


DEPARTMENT OF COMPUTER SCIENCE

CS Honours Project Final Paper 2024

Title: [*Advancing AI-Driven Molecular Generation: Exploring Decoding*]

Author: Gabriel Marcus

Project Abbreviation: Drug-GPT

Supervisor(s): Dr Jan Buys

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	5
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Advancing AI-Driven Molecular Generation: Exploring Decoding

Gabriel Marcus
University of Cape Town
Cape Town, South Africa
MRCGAB004@myuct.ac.za

ABSTRACT

Recent advancements in AI-driven molecular generation have attracted considerable interest within the pharmaceutical industry, particularly among companies engaged in drug discovery. The emergence of novel molecular line notations, such as Sequential Attention-based Fragment Embedding (SAFE), has further propelled this field forward. This study critically examines various decoding strategies within the framework of sequence models for molecular representations, an area that remains underexplored in current research. Utilising the pre-trained SAFE-GPT model, along with a smaller variant trained from scratch using the Moses Dataset, we systematically evaluate the impact of different decoding methods. Each experiment generates 10,000 molecules, with hyper-parameters iteratively adjusted to optimise performance. The generated molecules are assessed using metrics such as uniqueness, validity, and diversity. Our findings underscore the open-ended nature of molecular generation, which highlights the significance of constrained generation techniques in improving drug-likeness, as measured by the Quantitative Estimate of Drug-likeness (QED). These results indicate that careful selection of decoding strategies and the application of generation constraints can substantially enhance the quality and novelty of generated molecules, thereby contributing to more effective drug discovery processes. Moreover, we find that the large SAFE-GPT model demonstrates optimal performance when utilising temperature sampling during decoding, coupled with a repetition penalty. The repetition penalty discourages the model from repeating sequences to promote diversity. In contrast, the smaller SAFE-GPT model achieves its best performs with top-p sampling and without constraining it.

1 INTRODUCTION

Drug discovery has traditionally been an expensive and time-consuming process, that requires large teams to search through vast chemical spaces. These traditional approaches of finding a "lead compound", through methods like high throughput testing are limited in their ability to explore the extensive array of viable drug-like molecules [2]. Discovering new drugs requires that new compounds with pharmaceutically-suitable molecular properties are identified. However, progress is hindered by the extensive search space [9]. There have been attempts to use Artificial Intelligence (AI) to solve this problem. [6, 9]

The rise of AI, particularly machine learning, has yielded new strategies to aid in the discovery of new molecules [9]. In the context of drug discovery, generative AI models succeed by learning the underlying distribution of known molecules from their training data [9]. Once the model has learned this distribution, it can extrapolate and generate new, novel molecular structures that follow the same pattern. Molecular generative models can explore vast chemical spaces faster than traditional methods; these machine learning models learn the distribution of molecules by utilizing a "molecular language."

Previously, if one wanted to represent a molecule as a string, the Simplified Molecular Input Line Entry System (SMILES) [17] would be used. However, SMILES has several shortcomings, leading researchers to develop other molecular linear notations, such as the Sequential Attention-based Fragment Embedding (SAFE) format [9].

Molecular generation tasks can now be approached using sequence modelling methods which were originally developed in Natural language processing (NLP) . This means that molecular generation can be framed as a sequence generation task with the help of molecular line notations like SAFE or SMILES [9, 17]. This allows researchers to perform molecular generation tasks using a molecular language rather than relying on complex 3D generation techniques. The authors of SAFE trained a Generative Pre-trained Transformer (GPT) on SAFE data and achieved good results.

The transformer model [16] is considered state-of-the-art and has achieved high performance in NLP tasks. Transformer models excel compared to older NLP models due to their self-attention mechanism, which allows them to capture long-range dependencies and learn complex patterns present in molecules [16]. Despite the success of the transformer model, few studies have investigated the role that decoding strategies play in improving molecular generation.

Research in NLP has shown that one's choice of decoding algorithm can have significant impacts on the quality of generated text [5]. However different decoding methods have not been compared systematically for molecular generation, which suggests an opportunity to improve generation quality for this task [5, 15]. Greedy and beam search are evaluated as deterministic methods, while temperature and top-p sampling are evaluated as stochastic methods. Both deterministic and stochastic methods are used in different tasks.

In this study, we aim to explore various decoding strategies to determine the best approach for achieving high-quality, drug-like molecular generation using SAFE-GPT models [9]. The SAFE-GPT model has shown promising results in molecular generation tasks and demonstrates the potential of the SAFE molecular notation. To this end, we investigate:

(1) The impact of different decoding strategies, including stochastic and search-based methods, on key metrics of generated molecules, such as uniqueness, diversity, drug-likeness (QED), and validity, and whether the results are consistent for both large and small models.
(2) The effect of constraining decoders on the quality of the generated output in terms of the evaluation metrics.

2 BACKGROUND

2.1 Molecular Notation

2.1.1 SMILES. The SMILES molecular line notation has been the standard molecular line notation since the study in which it was developed was first published in 1988 [17]. SMILES has been the most widely adopted molecular line notation in chemo-informatics [9], because it simplified a complex 3D task into a more human-readable 2-D task, which could now be understood by a computer.

SMILES is a compact, human-readable string representation. SMILES encodes the structure of a molecule by using a combination of characters and rules for connectivity [17]. Smile does, however, have some shortcomings, and presents several issues. Molecules in the SMILES notation are not always unique, because a molecule can be written in several different ways using the SMILES notation. Several other flaws can result in invalid or unwanted molecules being formed; SMILES' lack of robustness has resulted in the creation of other molecular line notation languages like SAFE and SELFIES.[6].

2.1.2 SAFE. SAFE [9] was introduced to improve and fix the shortcomings of SMILES. SAFE is able to solve issues surrounding SMILES by re-imagining the SMILES strings as an unordered sequence of interconnected fragment blocks. SAFE's notational structure ensures that it remains a valid SMILES representation. [9]. SAFE therefore maintains compatibility with existing SMILES parses [9].

2.2 SAFE-GPT

SAFE-GPT [9] is a Transformer-based generative model specifically designed for molecular design tasks. It uses the SAFE molecular notation representation. It has been successful in generating pharmaceutically viable molecules. The SAFE notation encodes molecules as sequences of interconnected fragments. This allows SAFE-GPT to more effectively capture structural and chemical properties compared to traditional string-based representations such as SMILES. By encoding molecules in this manner, SAFE-GPT enhances both the controllability and interpretability of generated molecules relative to previous models based on SMILES notation[9]. SAFE-GPT is a decoder-only GPT-2-like model with task-specific adaptations. While SAFE-GPT has demonstrated impressive results in generating diverse and novel molecules, its performance, like that of any model, is heavily dependent on the quality and diversity of its training data. Further research is needed to fully understand the model's ability to generalise across chemical spaces.

2.3 Generation Tasks

In the field of NLP, tasks can be broadly categorised into two main types: open-ended generation, and directed generation.

2.3.1 Directed Generation. Directed generation tasks are defined through (input, output) pairs, where the output is a constrained transformation of the input [4]. It is also known as close-ended generation. An example of this would be a machine translation task, where the input would be an English sentence, and the output would be the sentence translated into German. Another example is text summarisation where the input is a document and the output is a concise summary of the document. Directed generation tasks typically favour deterministic methods [15], as these methods perform better in scenarios like summarisation where the output needs to be closely aligned with the input [15]. Repetition and genericness are less problematic in directed generation than in open-ended generation tasks [4]. This is due to the constrained nature of the tasks.

2.3.2 Open-Ended Generation. In open-ended generation, the input context restricts the space of acceptable output generations, but the total output space remains large. Unlike directed generation, which starts with a tightly scoped search space, open-ended generation allows for a significant degree of freedom in determining what can plausibly come next [4]. Examples of open-ended generation include conditional story generation and contextual text continuation. As a result, open-ended generation typically favours the use of stochastic methods [15]. The choice between stochastic and search-based decoding methods depends on whether the task is open-ended or directed.

2.4 Decoders

Decoding strategies fall into two broad categories: search based methods and stochastic based methods [8]. The selection of a decoding strategy depends on the specific requirements of the molecule generation task. Selecting the wrong type of decoder can yield poor results, while using an appropriate decoder results in high quality molecule generation[15]. The goal for optimally using a decoder is understanding the problem at hand and selecting the correct decoder for that task.

2.4.1 Greedy. Currently, there is no sub-exponential algorithm available for finding the optimal decoded sequence. In certain applications, a naive approach, such as the greedy algorithm, is appropriate. The greedy function, which employs arg-max, selects the token with the highest probability at each time interval [5, 15]. This approach often results in short and repetitive output sequences. Since the greedy algorithm always takes the 'greedy' approach, it does not explore multiple sampling options. Due to its simplicity, the greedy algorithm is rarely used in advanced language modelling [5]. Instead, models typically employ other decoding methods, such as beam search, which is commonly employed in directed generation tasks. The greedy algorithm can be described by the following iterative process:

$$x_i = \arg\max_{x_i} P(x_i | x_{<i}) \tag{1}$$

where this process is applied iteratively for each time step i to select the token with the highest probability at each step.

2.4.2 Beam search. Computing the most likely overall output sequence is an intractable problem[5]. The beam search decoding algorithm attempts to find the most probable output sequence by performing a breadth-first search over a restricted search space. This approach allows beam search to approximate finding the most likely sequence. The logic for the algorithm is as follows:

(1) At each decoding step, the algorithm maintains record of the W most probable hypotheses.
(2) The next set of partial hypotheses are chosen by expanding every path from the existing set of W hypotheses, and then choosing the W with the highest scores.[5, 15]

Due to the search spaces being large, beam search will only evaluates a subset of the overall search space. Beam searches produce multiple high-likelihood sequences which differ slightly [5].

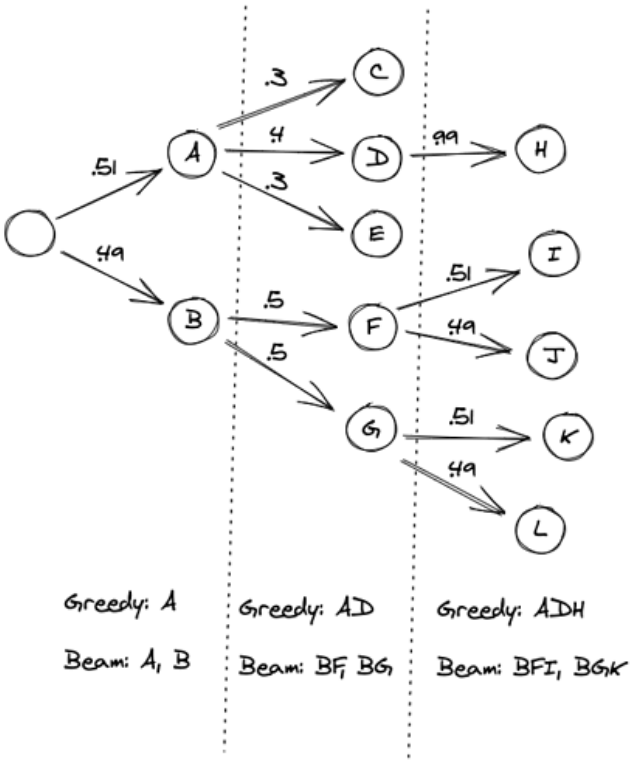


Figure 1: A visual representation of how beam and greedy decoding are performed [12].

Temperature sampling. Temperature sampling is a stochastic decoding strategy. It introduces randomness into the token selection procedure by sampling from the next-token probability distribution. The temperature hyper-parameter τ controls the flatness of the distribution, which allows for a trade-off between the quality and diversity of the generated molecules [11, 15]. Higher temperature lead to more uniform distributions, which encourages the generation of diverse molecules. Lower temperature parameters focus on high-probability tokens, resulting in molecules that are more likely to resemble the training data and be pharmaceutically-valid.

The probability of selecting a token x_i given the prefix $x_{<i}$ is given by:

$$P(x_i|x_{<i}) = \frac{e^{z_i/\tau}}{\sum_{j=1}^{V} e^{z_j/\tau}}$$

where:

V is the size of the vocabulary, i.e., the number of possible tokens. z_i is the logit (raw output score) corresponding to token x_i. The equation combines the softmax function with the temperature parameter. This results in the logits being transformed into a probability distribution. Thus a higher temperature flattens the distribution promoting diversity and a lower temperature prioritises higher probability tokens. After the equation is applied tokens are sampled accordingly [11].

2.5 Nucleus Sampling

Another stochastic-based sampling method is nucleus sampling, also known as top-p sampling. Nucleus sampling addresses some of the limitations [4] of temperature sampling. It does this by dynamically selecting a subset of the most probable tokens that account for a specified probability mass p [15]. Unlike temperature sampling, which samples from the entire vocabulary and can lead to either overly random or overly deterministic outcomes depending on the temperature setting, nucleus sampling focuses only on the most relevant tokens by truncating the probability distribution at the tail. This means that tokens contributing minimally to the overall probability mass are excluded from consideration, effectively eliminating the least likely tokens [4]

Nucleus sampling thus reduces the chances of generating irrelevant or nonsensical outputs like pharmaceutically-invalid molecules. This targeted approach allows nucleus sampling to balance diversity and coherence more effectively than temperature sampling, by ensuring that the generated sequences are both meaningful and varied. Mathematically, nucleus sampling can be explained as selecting the smallest set of tokens whose cumulative probability exceeds the specified threshold p, ensuring that only the most probable and contextually appropriate tokens are considered. Given a probability distribution $P(x|x_{1:i-1})$ over the vocabulary V, nucleus sampling first defines the top-p vocabulary $V^{(p)} \subseteq V$ as the smallest set of tokens such that:

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p.$$

$$P'(x|x_{1:i-1}) = \begin{cases} \frac{P(x|x_{1:i-1})}{p'} & \text{if } x \in V^{(p)}, \\ 0 & \text{otherwise.} \end{cases}$$

The next token is then sampled from the re-scaled distribution P'. There are other approaches one can make use of to generate high quality outputs, such as constraining decoders generation.

2.6 Constraining

Constraining the decoder's output guides generation to produce molecules with higher pharmaceutical applicability [2, 14].

This study looks at repetition based constraint. A repetition-based constraint prevents the model from recycling sequences. The repetition based-constraint is implemented through a repetition penalty mechanism. The repetition penalty works by keeping track of the frequency of every token that has been outputted. Before the model selects the next token, the repetition penalty applies a penalty to each token in the sampling distribution, with the magnitude of the penalty proportional to the token's previous occurrence frequency. The model will then sample from the adjusted distribution. The penalty applied to a particular token increases every time the token is sampled. The higher the repetition penalty hyperparameter, the stronger it will discourage repetition [10].

The choice of molecular representation can impact the effectiveness of constraint techniques and the quality of the generated molecules. The study will evaluate these techniques using the aforementioned evaluation metrics.

2.7 Evaluation Metrics

Evaluating the viability and quality of machine-generated molecules is key in developing future generative models for molecular design. Several metrics have been proposed to assess the performance of machine generated molecules, each focusing on different aspects of the generated molecules.

2.7.1 Validity and Uniqueness. These metrics are seen as the most important metrics when evaluating generative models. Validity measures the percentage of chemically valid structures according to a molecular parsing tool such as RDKit [7].

Given a set of generated molecules G, the validity is defined as:

$$\text{Validity}(G) = \frac{|m \in G : \text{is\_valid}(m)|}{|G|} \times 100 \tag{2}$$

where is_valid$(\cdot)$ is a function that determines the chemical validity of a molecule using a molecular parsing tool,like RDkit [10].

Uniqueness [10] is the percentage of non-duplicate molecules within a set of generated compounds. It is calculated as:

$$\text{Uniqueness}(G) = \frac{|\text{unique}(G)|}{|G|} \times 100 \tag{3}$$

where unique$(\cdot)$ returns the set of unique molecules in G [10].

These metrics provide a basic assessment of a generative model's performance and have been widely reported in the literature. It must, however, be noted that these metrics do not *directly* determine the quality or desirability of generated molecules for specific tasks; these metrics determine only their basic properties.

2.7.2 Diversity. 'Diversity' describes the chemical heterogeneity of generated molecules. Diversity is quantified using various metrics; one of these is the Tanimoto similarity between the generated molecules and the training data. Another key metric is the distribution of molecular properties like molecular weight, logP, and synthetic accessibility score.

This study measured diversity by calculating the average pairwise Tanimoto distance between the generated molecules based on their fingerprint representations [13]. Given a set of generated molecules G and a fingerprint function $f(\cdot)$, the diversity is calculated as:

$$\text{Diversity}(G) = \frac{2}{|G|(|G|-1)} \sum_{i=1}^{|G|} \sum_{j=i+1}^{|G|} \text{Tanimoto}(f(m_i), f(m_j)), \tag{4}$$

where $m_i, m_j \in G$ and Tanimoto$(\cdot,\cdot)$ is the Tanimoto similarity between two fingerprint representations [10].

A higher diversity score indicates that the generated molecules are more chemically diverse and cover a wider range of the chemical space. However, this diversity metric has limitations in fully capturing the scope of chemical diversity, as it depends on the choice of fingerprint representation and may not account for all the relevant structural and functional differences between molecules.

2.7.3 Quantitative Estimate of Drug-likeness (QED). The Quantitative Estimate of Drug-likeness (QED) [1] is a metric that assesses the drug-likeness of a molecule based on its physicochemical properties. It combines eight properties (molecular weight, logP, hydrogen bond donors, hydrogen bond acceptors, polar surface area, rotatable bonds, aromatic rings, and structural alerts) into a single score ranging from 0 (low drug-likeness) to 1 (high drug-likeness).

The QED score is calculated using a weighted geometric mean of the individual property scores:

$$\text{QED} = \left(\prod_{i=1}^{n} d(p_i)^{w_i} \right)^{\frac{1}{\sum_{i=1}^{n} w_i}}, \tag{5}$$

where $d(p_i)$ is the desirability score for property p_i, w_i is the weight assigned to property p_i, and n is the number of properties [1].

QED provides a quantitative estimate of a molecule's drug-likeness and can be used to prioritise molecules that possess pharmaceutically useful properties.

2.7.4 Synthetic Accessibility Score. The Synthetic Accessibility (SA) Score [3, 10] is a metric that is used to approximate the ease of synthesis of a molecule based on the complexity of its substructures. SA is calculated by the summation of the complexity contributions of every atom in the molecule, along with additional factors such as the presence of chiral centres and the complexity of the molecular graph.

The SA score of a molecule m is defined as:

$$\text{SA}(m) = \frac{1}{n} \sum_{i=1}^{n} c_i + \frac{1}{n} \sum_{i=1}^{n} r_i + s + t, \tag{6}$$

where n is the number of atoms in the molecule, c_i is the complexity of the i-th atom's substructure, r_i is the number of rings in the substructure, s is the overall complexity of the molecular graph, and t is the number of chiral centres [3] .

SA ranges from 1 (easy to synthesis) to 10 (difficult to synthesise). Incorporating SA into the evaluation pipeline can help prioritise generated molecules that are more feasible to synthesise, thus increasing the practical utility of the generative models. However, the SA score is only an approximation of synthetic accessibility and may not capture all the nuances and challenges involved In real world chemical synthesis, such as reagent compatibility, reaction conditions, and purification steps. Validity, uniqueness, diversity, QED, and SA remain widely used evaluation metrics for assessing the performance of molecular generative models, despite these limitations.

2.8 Model Architectures

The SAFE-GPT model, which is available on Hugging Face [1], was used as a foundation for the research in this study. The code provided by the authors served as the starting point for implementing a smaller variant of the SAFE-GPT model that was subsequently trained for this study. The architecture of the SAFE-GPT models used in this study is as follows:

- **SAFE-GPT-20M**: 6 layers, 8 attention heads per layer, hidden state size of 768, approximately 20 million parameters.
- **SAFE-GPT**: 12 layers, 12 attention heads per layer, hidden state size of 768, approximately 60 million parameters.

[1] https://huggingface.co/datamol-io/safe-gpt

3 METHODOLOGY

The evaluation of decoding methods was conducted using the SAFE library, which is a wrapper for the Hugging Face Transformer library. The experiments were carried out on the university's computer system called BadKamer [2].

The experimentation process can be divided into two main phases, each with nested sub-phases.

3.1 Phase 1: Evaluating General Decoding Methods

In this phase, we evaluated general decoding methods for two models: the large SAFE model and a smaller model trained from scratch. This involved running various decoders on each model and iterating through their possible hyperparameters.

3.2 Phase 2: Constraining Decoding

The second phase involved constraining the decoding process for each architecture using the technique previously discussed.

After completing both sets of experiments, we compared the metrics of the different decoders as applied to each respective model. The large model was acquired from the SAFE library [3], and the reference molecules were retrieved from the MOSES repository [4].

Given that this study looks at molecular generation by making use of sequence models, domain-specific metrics were implemented. The metrics were evaluated individually, and the results were discussed both separately and collectively.

3.3 Datasets

3.3.1 MOSES Dataset. The Molecular Sets (MOSES) dataset [10] was used in the experimentation procedure to provide analysis of the model's performance on a smaller, drug-discovery-focused dataset. The MOSES dataset contains 1.9 million drug-like molecules. It is represented using the SMILES format and is refined from the ZINC Clean Leads collection. It filtered out roughly 2.5 million zinc molecules. The dataset used to train this study's smaller SAFE-GPT model was the same as the dataset used to train the smaller SAFE-GPT model in the original SAFE study. This dataset was used to allow for the replication and expansion of the original study's findings. As the molecules are encoded in the SMILES representation, this dataset will be converted to the SAFE representation by applying the SAFE algorithm [9].

3.4 Additional Resources

This study required only moderate computational resources, because this task involves inference and evaluation. However, evaluating and experimenting with the various decoding strategies will have varying computational requirements.. We optimised the use of available computational resources to ensure the timely completion of all experiments. Inference was conducted using UCT's Badkamer computer system.

[2]Badkamer consists of 2 Nvidia RTX4090 series
[3]Accessible through: https://safe-docs.datamol.io/stable/
[4]Moses Dataset url: https://github.com/molecularsets/moses

4 RESULTS AND DISCUSSION

Molecular generation is a technical problem in which numerous metrics determine the quality of a generated molecule. Evaluating the output of a molecular generation task requires that multiple factors be considered. A comprehensive interpretation of results can only be achieved by examining all metrics collectively. This study replicated the original SAFE papers results for both the large and small model. The hyper-parameters for the original authors, 'default' model are a top-k of 50, top-p of 1 and temperature of 1. A top-p of 1 means it will look at all the token in the distribution and a temperature of 1 effectively means no modifications are applied to the logits, resulting in the model sampling from the original distribution. This study made use of a repetition penalty of 1.2 when applying a repetition penalty. Although as shown in figure 3 various penalties could have been used.

4.1 Large model

4.1.1 QED. In analysing search-based decoding strategies, we observe the following: Greedy decoding results in a low QED of 0.053, while beam search yields a slightly higher, yet still low, QED of 0.076.

For the default model a QED of 0.69 was achieved and when a repetition penalty was added it achieved an increase to 0.699.

For top-p sampling, QED increases with the hyper-parameter value. The maximum QED for top-p is observed at a hyper-parameter value of 0.8, where it reaches approximately 0.7. The top-p QED graph exhibits a monotonic upward trend as the hyper-parameter increases. Moreover, as shown in Figure 2, constraining top-p decoding appears to enhance QED, though as the hyper-parameter increases, the constrained and unconstrained results tend to converge.

The maximum QED for temperature sampling occurs at a hyper-parameter of 0.8, with a QED value of 0.7. The QED graph for temperature follows a parabolic shape, as depicted in Figure 2. Constrained temperature sampling, especially in the large SAFE-GPT model, improves QED for low hyper-parameters, but as the hyper-parameter value increases, performance degrades.

4.1.2 SA score. The SA score for greedy decoding is low at 2.77, whereas beam search yields a relatively high SA score of 4.45. The default model had an SA score of 0.993 and when constrained with a repetition penalty increased to 3.847. For top-p sampling, the SA score increases with higher hyper-parameter values. Comparing the constrained and unconstrained models, the constrained top-p decoding results in an overall higher SA score, as shown in Figure 2.

The SAS score is 3.26 at the hyper-parameter value where maximum QED is achieved for temperature sampling. As the hyper-parameter value increases, so does SA score. The constrained temperature model consistently outperforms the unconstrained version in terms of SA, as illustrated in Figure 2.

4.1.3 Diversity. The greedy algorithm produces a diversity score of 0, as it always optimises for the best molecule. In contrast, beam search achieves a diversity score of 0.47. For the default model a diversity of 0.868 was achieved and when constrained it increased to 0.881.

Table 1: QED best performance for each decoding strategy large SAFE-GPT model

Decoder	Hyper-parameter	QED	Validity	SA score	Diversity
Default settings	N/A	0.690	0.993	3.417	0.868
Default settings with repetition penalty	N/A	0.699	0.939	3.847	0.881
Greedy	N/A	0.053	1.000	2.768	0.000
Beam search	20	0.131	0.500	3.326	0.537
Beam search with repetition penalty	20	0.131	0.500	3.326	0.537
Temperature	0.8	0.701	0.940	3.258	0.871
Temperature with repetition penalty	0.6	0.740	0.969	3.582	0.866
Top-p	0.8	0.720	0.947	3.150	0.864
Top-p with repetition penalty	0.8	0.748	0.967	3.624	0.866

Table 2: QED best performance for each decoding strategy small SAFE-GPT model

Decoder	Hyper-parameter	QED	Validity	SA score	Diversity
Default settings	N/A	0.801	0.992	2.494	0.865
Default settings with repetition penalty	N/A	0.790	0.984	2.734	0.873
Greedy	N/A	0.874	1.000	1.42	0.000
Beam search	40	0.876	1.000	2.187	0.568
Beam search with repetition penalty	20	0.130	0.500	3.320	0.537
temperature	0.5	0.833	1.000	2.100	0.823
Temperature with repetition penalty	0.6	0.816	0.999	2.480	0.859
Top-p	0.8	0.834	1.000	2.245	0.842
Top-p with repetition penalty	0.8	0.816	1.000	2.543	0.861

For top-p sampling, diversity follows a logarithmic pattern; it increases rapidly and then plateaus. At the hyper-parameter value of 0.8, which maximises QED, top-p achieves a diversity score above 0.75. As seen in Figure2, constraining top-p sampling enhances Diversity for lower hyper-parameter values.

Similarly, the diversity score for temperature sampling follows a logarithmic trend. At a temperature of 1, Diversity exceeds 0.85 for both constrained and unconstrained models. As depicted in Figure 2, the constrained model initially performs better with lower hyper-parameter values, but as the hyper-parameter increases, the constrained and unconstrained results converge.

4.1.4 Uniqueness and Validity. The validity for the default model was 0.993 which reduced to 0.939 when constrained with the repetition penalty. For top-p sampling, the large model consistently achieves high validity across all hyper-parameter values it does not fall below 0.9. In contrast, temperature sampling performs well up to a hyper-parameter of 1.25, after which the validity rapidly declines. The uniqueness begins low and increases as the hyper-parameter values increase for both temperature and top-p sampling.

4.2 Small model

For the small model, we observe that the general trendy for QED, Validity, SA and Diversity align with the large model.

4.2.1 QED. Unlike the large model, the small model achieves better results for lower hyper-parameters. The differences in the QED graph between top-p with the repetition constraint, and top-p without the repetition constraint, is less stark than in the large model. It performed better without the repetition constraint, unlike in the case of the large model. The temperature sampling behaved differently in the small model in comparison to the large model, where it continuously decreased as the hyper-parameter was raised. It also performed better without the repetition constraint. The QED has approximately 0.1 increase for both constrained and unconstrasined implementations.

4.2.2 SA. The SA scores in the small model produced a similar graph to the large model, but had lower SA scores overall. The small model likewise featured higher SA score for constrained versions.

4.2.3 Diversity. Diversity was almost identical to the large model; however, for temperature, the small model did not begin with low diversity, which distinguished it from the large model.

4.2.4 Uniqueness and Validity. The small model featured high validity, which it had in common with the large model. The uniqueness follows the similarly from the large model.

5 DISCUSSION

5.1 Search vs sampling based

Sampling methods outperformed search-based methods in this study. Notably, search-based methods yielded lower SA scores, indicating that they generate molecules that are potentially easier to synthesise. However, due to their inherent design, search-based methods produced molecules with poor QED and diversity. In contrast, sampling-based methods demonstrated superior overall performance; this was likely due to the task's creative nature rather than its directed characteristics.

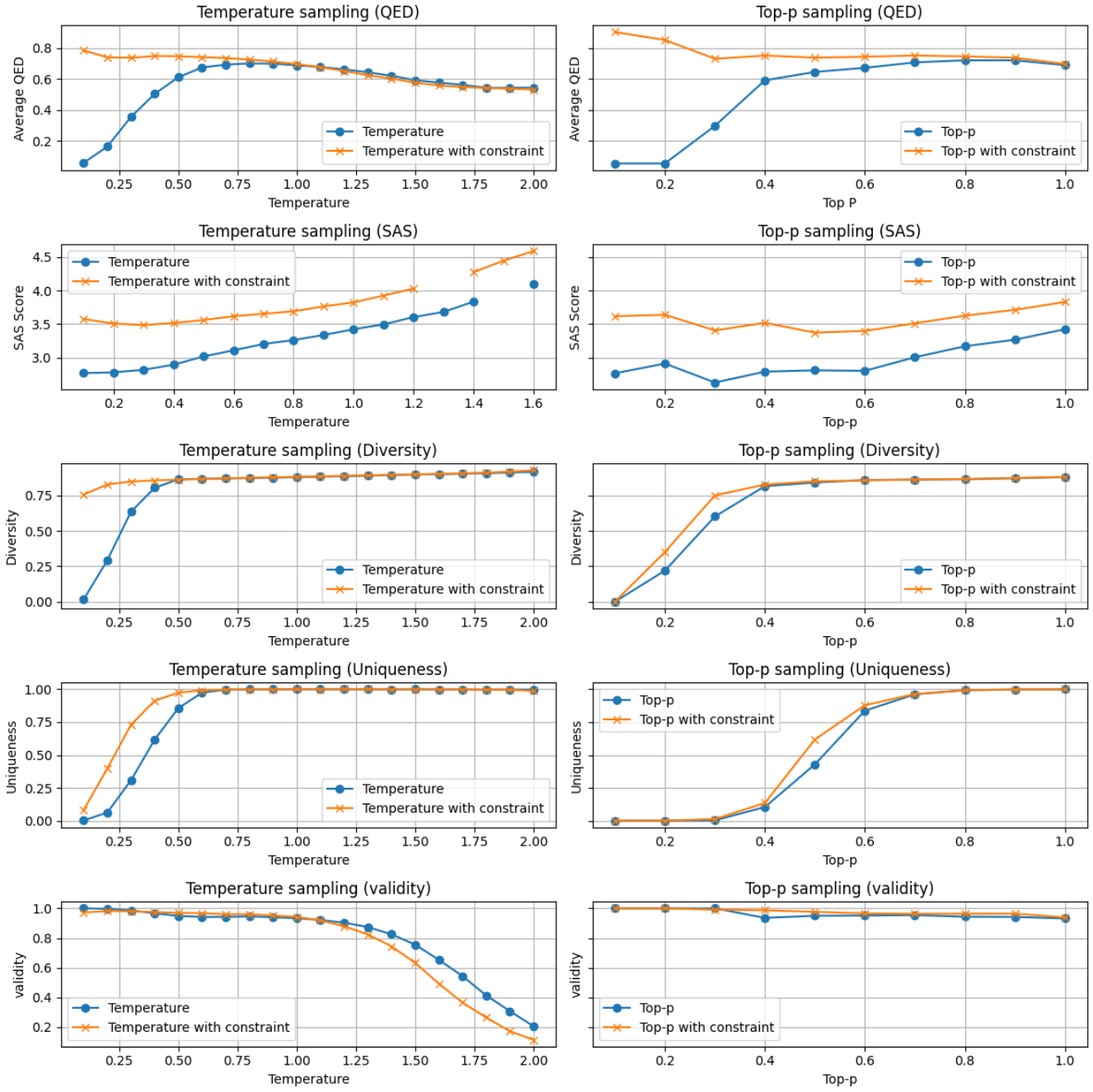


Figure 2: Evaluation results of various decoding strategies for the Large SAFE-GPT model over the various hyper-parameters

5.2 Temperature

5.2.1 Big Model. Temperature sampling outperformed the default model, achieving a higher QED, lower SA and higher diversity. The temperature method with a repetition penalty also achieved a significantly higher QED score compared to both the default model and the default model with the repetition penalty. Temperature sampling generated highly diverse, valid molecules with high QED scores. The constrained version of the decoder yielded similar results. The algorithm promoted diversity effectively, achieving optimal outcomes with a hyper-parameter value of approximately 1. The model performs exceptionally well with low hyper-parameter values when constraints are applied. Applying constraints appears to be initially effective, but as the sampling space expands, its positive effects diminish. A similar trend is observed in the large model

for top-p sampling. When this occurs, it also leads to an increase in the SA score. These findings suggest that these tasks benefit significantly from restricting search space with implemented constraints.

5.2.2 Small model. The temperature method used in the small model also outperformed the default model in QED, Validity and SA. The small model demonstrated good performance under temperature sampling. However, the fact that the QED value decreased over time suggests that the model may not have fully learned the underlying distribution. When presented with a larger sampling area, the model struggled to sample effectively. Constrained temperature sampling proved detrimental, resulting in reduced QED scores, increased SA scores, and only negligible improvements in diversity.

5.3 Top-p

5.3.1 Big model. The top-p sampling method achieved a higher QED, lower SA score compared to the default model where the constrained top-p sampling achieved notably higher QED score compared to the default model. Constraining the model improves QED for low to mid-upper hyper-parameter values. However, this improvement comes at the cost of significantly increased SA scores.

5.3.2 Small model. The small model making use of top-p achieved a higher QED and validity. It also achieved a lower SA score. The same applies to the constrained top-p. As illustrated in Figure 4, constraining generally has a negative impact on QED scores. While constraining by repetition marginally enhances diversity, it is crucial to note that SA scores deteriorate when constraints are applied.

5.4 Limitations

This study was challenged by a lack of domain knowledge, which slowed down experimentation due to unfamiliarity with the topic at hand. This effect was partially mitigated by consulting with relevant professionals.

6 CONCLUSIONS

6.1 Experimentation Insights

Our experimentation revealed several key findings. Firstly, it was found that this task is an open-ended generation task. This makes sampling-based methods more suitable than search-based approaches. A notable contrast in performance metrics emerged between these search and sampling-based methods. Sampling-based methods consistently outperformed search-based methods across all metrics except with the small model's QED score, where search-based methods performed comparably, but demonstrated poor diversity.

Moreover, under temperature sampling, the generated output quality begins to degrade when the hyper-parameter exceeds 1 for both the small and large models, suggesting that while creativity is essential for this task, it requires some form of guidance. Interestingly, adding a repetition penalty improved performance only for the large model.

Search-based methods often produced high SA scores alongside low QED scores. Furthermore top-p sampling required a high hyper-parameter to match the uniqueness score of the default model, while temperature sampling generated unique molecules with a much lower hyper-parameter. Moreover, top-p consistently resulted in high validity, whereas the validity of temperature sampling declined once the hyper-parameter exceeded 1.

6.2 Impact of Constraining on Model Output

Regarding the impact of constraining on model output, our analysis revealed the following: The effectiveness of different constraining mechanisms varies depending on the specific task. Constraining methods can potentially enhance QED and diversity while reducing SA scores, contingent on the model employed. Constraining could be performed if one uses the large model, as it achieves higher results compared to the unconstrained model, though it will increase the SA score.

This paper recommends, if one were to use the large model, that they should make use of temperature sampling with a hyper-parameter of 0.8 alongside a repetition penalty. If one were to use the small model, the results suggest it is best to use top-p sampling with a hyper-parameter of 0.8.

6.3 Implications and Contributions

Our research demonstrates that decoding strategies significantly influence the quality of the model's generated output. The impact of constraining varies depending on the model, potentially yielding either beneficial or detrimental effects. This study provides valuable insights into various decoding methods in the field of molecular generation, highlighting alternatives to default generation techniques. Specifically, we offer guidance on selecting appropriate decoders for *de novo* generation based on desired outcomes, such as high QED or low SA scores.

6.4 Future Directions

Future research should focus on examining decoders behaviour in other molecular generation tasks. In addition, further exploration can be done on constraining methods to enhance the QED and diversity as well as reduce the SA score. Moreover one can investigate why constraining significantly improved QED performance for low hyper-parameters in the larger model.

REFERENCES

[1] G Richard Bickerton, Gaia V Paolini, J'er'emy Besnard, Sorel Muresan, and Andrew L Hopkins. 2012. Quantifying the chemical beauty of drugs. *Nature chemistry* 4, 2 (2012), 90–98.

[2] Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. 2019. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design Engineering* 4, 4 (2019), 828–849.

[3] Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* 1, 1 (2009), 1–11.

[4] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).

[5] David Ippolito, Rene Kriz, Masha Kustikova, João Sedoc, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. *arXiv preprint arXiv:1906.06362* (2019).

[6] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. 2020.

[7] Greg Landrum et al. 2023. RDKit: Open-source cheminformatics. *Online. http://www. rdkit. org* (2023).

[8] Xiaolei Li, Ari Holtzman, David Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097* (2022).

[9] Elaheh Noutahi, Chiara Gabellini, Matthew Craig, Jia Siang Lim, and Pulchérie Tossou. 2023. Gotta be SAFE: A New Framework for Molecular Design. *arXiv preprint arXiv:2310.10773* (2023).

[10] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. 2020. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. arXiv:1811.12823 [cs.LG]

[11] M. Renze and E. Guven. 2024. The Effect of Sampling Temperature on Problem Solving in Large Language Models. (2024). arXiv:2402.05201 [cs.CL]

[12] Rob. 2020. Is beam search always better than greedy search? https://discuss.huggingface.co/t/is-beam-search-always-better-than-greedy-search/2943. Accessed: 2024-08-17.

[13] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* 50, 5 (2010), 742–754.

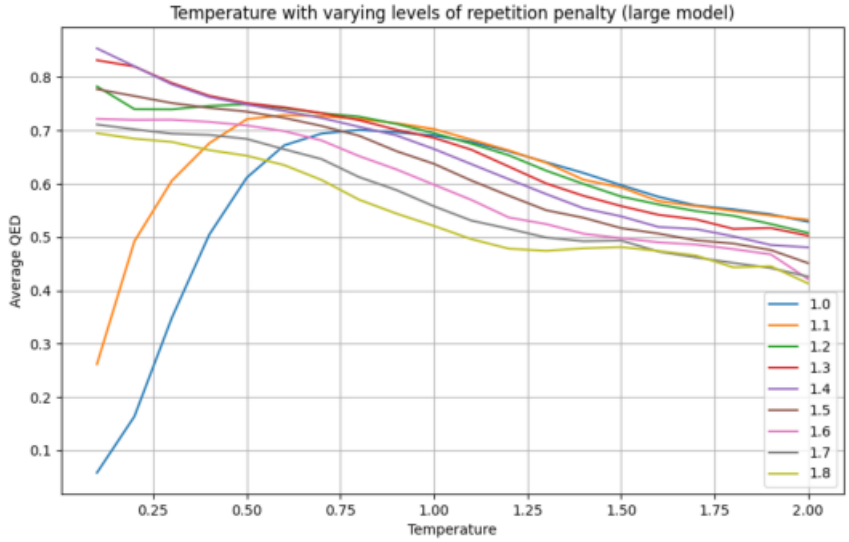
[14] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. 2018. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* 361, 6400 (2018), 360–365.

[15] Chuan Shi, Han Yang, Deng Cai, Zhan Zhang, Yan Wang, Yiming Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of llms. *arXiv preprint arXiv:2402.06925* (2024).

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[17] David Weininger. 1988. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28, 1 (1988), 31–36.

A APPENDIX



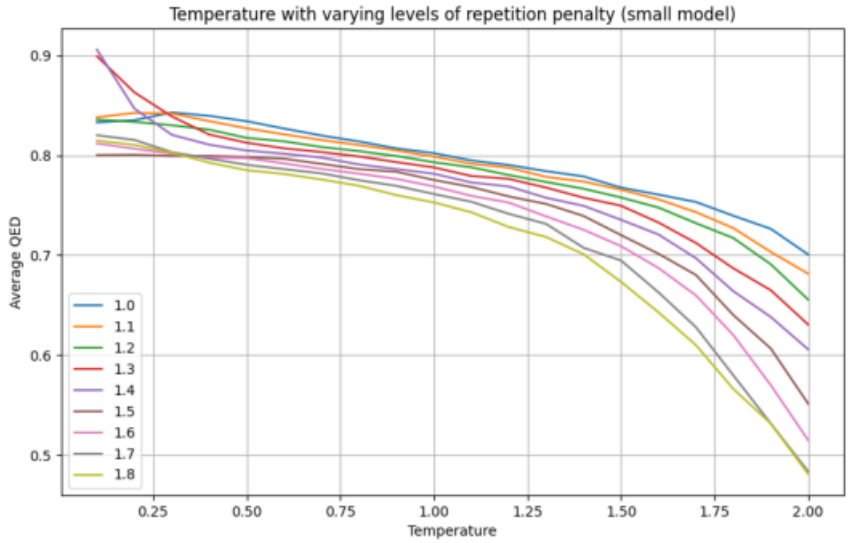


Figure 3: Repetition penalty across temperature hyper-parameters

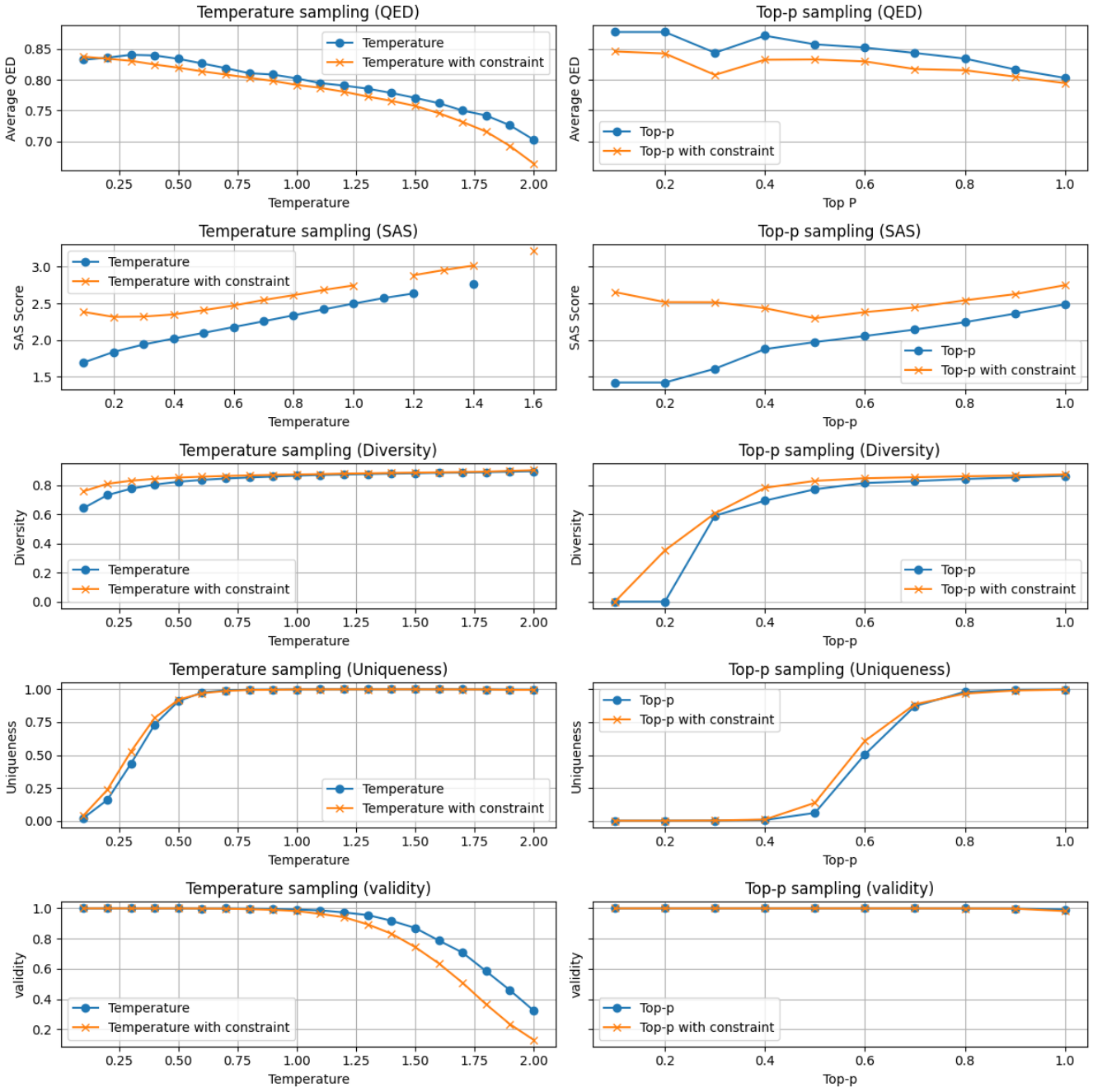


Figure 4: Evaluation results of various decoding strategies for the Small SAFE-GPT model over the various hyper-parameters