Grammar Control for Language Learning with Sequential Monte Carlo and Small Language Models

Kate McCurdy[1] **Jacob Hoover Vigly**[2] **Thomas Hikaru Clark**[3]
Karolina Stańczak[4] **Yahya Emara**[5] **Joshua B. Tenenbaum**[2,3]
Alexander K. Lew[6] **Ryan Cotterell**[4] **Timothy J. O'Donnell**[2,7,8,9]
[1]University of Göttingen [2]CHI-FRO [3]MIT [4]ETH Zürich [5]Cornell University
[6]Yale University [7]McGill University [8]Mila [9]Canada CIFAR AI Chair

Abstract

Second language learners benefit from exposure to level-appropriate grammar in interactive language use. While prompted frontier language models can usually produce target grammar structures, less resource-intensive models that can be run on device often fail in this task. We use Sequential Monte Carlo (SMC) to steer dialogue turn generation in small language models toward target grammar constructions from an expert-developed pedagogical resource. With a minimal set of hand-coded potential functions, SMC achieves near-perfect coverage of target grammar, enabling small models to outperform state-of-the-art results from models 140 times larger. We also use code synthesis with frontier models to generate new potentials, showing that our method scales to hundreds of target grammar skills with no need for additional training or supervised data.

1 Introduction

A key task in second language pedagogy is exposing language learners to level-appropriate target grammatical structures in diverse natural contexts. For example, a learner who speaks English at an advanced beginner level (e.g., level A2 in CEFR, the Common European Framework Reference) should be able to understand and use the verb *enjoy* followed by a verb in its gerund (*-ing*) form (e.g., *enjoy swimming*, *enjoy cooking*). A capable English-language instructor can use this target construction in dialogue (e.g., *I enjoy cycling. What exercises do you enjoy?*), encouraging learners to practice and develop this grammatical skill — that is, producing grammatical constructions (e.g., *enjoy swimming*) rather than ungrammatical word sequences (e.g., *enjoy to swim*). We would like language models to provide similar interactive support for learners. Whether a model can do so reliably, however, depends heavily on its scale. Frontier language models can usually produce learner-appropriate grammatical constructions through prompting, but smaller, less resource-intensive models often fail to reliably produce grammar targets (Sun et al., 2023; Ribeiro-Flucht et al., 2025). Bridging this gap could make efficient, on-device language learning tools more feasible, reducing costs and reliance on cloud-based frontier-model APIs while broadening access to learning applications.

This problem is naturally formulated as a constrained generation task: Given some set of level-appropriate grammar skills, we want to constrain language model generations such that they contain the target constructions. Pedagogical resources such as the English Grammar Profile (O'Keeffe and Mark, 2017) identify a range of grammatical targets at specific proficiency levels. Previous research has investigated grammar control via prompting, fine-tuning, reinforcement learning, and classifier-based constrained decoding, but none have yet achieved reliable elicitation of target constructions from small language models (Okano et al., 2023; Glandorf and Meurers, 2024; Glandorf et al., 2025).

We can formulate constrained generation as a probabilistic inference problem. A language model defines a prior $p(\boldsymbol{x})$ over strings. Given a *potential* $\varphi(\boldsymbol{x})$ — a binary function that checks whether a string $\boldsymbol{x}$ satisfies the constraint — we want to sample from the posterior $\frac{p(\boldsymbol{x})\varphi(\boldsymbol{x})}{\sum_{\boldsymbol{x}'} p(\boldsymbol{x}')\varphi(\boldsymbol{x}')}$. The denominator term sums over all strings, which is intractable; this motivates use of approximate inference techniques such as Sequential Monte Carlo (SMC). SMC builds an importance-weighted population of candidate strings (*particles*) whose empirical distribution approximates the target posterior. Recent work has used SMC to approximate target distributions for constrained generation from language models (Lew et al., 2023; Zhao et al., 2024; Loula et al., 2025; Lipkin et al., 2025; Xefteri et al., 2025; Grand et al., 2025; Emara et al., 2026; Golowich et al., 2026; Kim et al., 2025). Cru-

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 35943–35967
October 24-29, 2026 ©2026 Association for Computational Linguistics

cially, SMC operates at inference time on any base language model. In our setting, it enables small, efficient models to achieve high levels of target grammar coverage without additional training.

In this work, we specify potential functions via the GenLM Control library,[1] using short, interpretable, human-readable programs to steer generation. For example, a potential function for the *enjoy* + gerund construction might parse the candidate sentence and return a high score if and only if the verb *enjoy* is followed by a verb gerund, excluding superficially similar sequences (e.g. *enjoy something*). Because potential functions can be implemented as explicit programs, they can be flexibly developed, modified, and composed. Programmers can directly write functions to specify target grammar constructions, which we do in Experiment 1. Such programs can also be synthesized by large language models, and refined through iterative generation and testing, a technique known as self-steering SMC (Grand et al., 2025); we demonstrate that this approach can scale to cover tens (Experiment 2) or hundreds (Experiment 3) of targets. In all cases, potential functions can be inspected and refined by domain experts (e.g. instructors or curriculum designers), allowing flexible adaptation for specific pedagogical goals.

In this work, we apply SMC[2] to elicit dialogue turns with target grammar constructions from language models with fewer than 10 billion parameters (Figure 1). We make the following contributions:

- We show that SMC steering can raise the target grammar accuracy of a model with 0.5B parameters from 58% up to 98% — outperforming previous state-of-the-art (SotA) from a model 140 times larger — while optionally composing with other pedagogically-motivated constraints (Experiment 1).

- We show that frontier models can effectively synthesize grammar potential functions through self-steering, and combining up to 6 generated potentials at inference time outperforms previous SotA results (Experiment 2). Moreover, this approach scales to hundreds of targets with no need for supervised data or additional training (Experiment 3).

[1] genlm.org/genlm-control

[2] Code: github.com/kmccurdy/SMC-L2-grammar

ID (CEFR)	Can-do statement, describing grammar skill	Examples of correct target use in English learner corpora
258 (A1)	FORM: PHRASE + PHRASE Can use single word conjunctions (`'and, but, or'`) to combine phrases.	My home is very light and very comfortable. I live with my mother and her boyfriend. My house is small but very nice.
262 (A1)	USE: `'BUT'`, CONCESSIVE Can use `'but'` to add an unexpected contrast.	I really like my home, it is very small but it is very easy to clean. I like my home, it's small but so comfortable.
265 (B1)	FORM/USE: `'PLUS'` WITH CLAUSES Can use `'plus'` to connect clauses and sentences, often to point out a positive addition or advantage.	I would be more than pleased to receive you at home, it's not too big, but we could have a lot of fun, plus you won't have to pay a thing! It will be fun because everyone is going to be there. Plus, the park is the best place for a picnic, we can run and just have fun!

Table 1: Example EGP target grammar skills.

2 Background

Previous research on dialogue adaptation for second language learning has investigated learner-appropriate constraints on vocabulary (Tyen et al., 2022; Qian et al., 2023) and certain grammar categories (Okano et al., 2023). More recently, Ribeiro-Flucht et al. (2025) and Glandorf et al. (2025) explore controlled dialogue generation with a wider range of target grammar constructions grounded in the pedagogical framework of the English Grammar Profile (EGP), and establish SotA performance for specific learner-appropriate grammar targets. As our experiments replicate their setup, we first introduce the EGP, and then review these two recent studies in greater depth.

2.1 Targets: English Grammar Profile

We use the English Grammar Profile (EGP; O'Keeffe and Mark, 2017), a pedagogical resource that lists 1,222 grammar skills for learners of English at specific proficiency levels. This resource was developed through analysis of learner corpora, and identifies each grammar skill with the proficiency level in the Common European Framework Reference (CEFR) at which it is typically mastered. A skill is considered mastered when learners correctly produce the target grammar construction in appropriate contexts.

EGP grammar skills are expressed as "can do" statements with supporting examples, and may target either a specific grammatical *form*, its *use* in a particular discourse context, or both.[3] Table 1 gives some example skills from the category "coordinating conjunctions." Note that, while target

[3] For more details, see the EGP website: englishprofile.org

Prompt style	Prompt
B (Basic)	You are a friendly English tutor. I want to practice by 'Discussing enjoyable hobbies'.
T (Template)	I am a beginner English student. I want to practice patterns with to and -ing by Discussing hobbies with 'enjoy + ing'. I want to ensure that I can use 'enjoy' followed by the '-ing' form. For example, 'I enjoy eating, I enjoy doing, I enjoy going…'. Let's do a Q&A exercise. You (the AI) are a friendly English tutor. Tell me What you enjoy doing. Then ask me the same with short questions. ('I enjoy reading. What do you enjoy doing?', 'When I go out I enjoy dancing. How about you?')

Table 2: Partial basic vs. template prompts for EGP skill 1082 (*enjoy + -ing*) with the topic "hobbies" (Ribeiro-Flucht et al., 2025). For complete prompts and further details, see appendix B.

grammar constructions are expressed clearly, some skills are open to ambiguity in interpretation; for instance, evaluators of EGP ID 262 may differ in judgments of an "unexpected contrast." While we aim for consistent evaluation, such ambiguities lead to unavoidable annotator variation and noise.

2.2 Prior work

EGP skills have been used to define target grammar constructions in two recent studies establishing SotA for pedagogical grammar control, which our experiments build upon.

Ribeiro-Flucht et al. (2025) assess tutoring dialogue scenarios in which the language model is prompted to use a target grammar construction while engaging an L2 English student in dialogue on a specific topic. They identify 5 grammar skills at proficiency level A2, and test each skill with different topics. The authors compare two prompting approaches: a **basic (B)** prompt with relatively minimal instructions, and a more elaborate **template**-based **(T)** prompt with detailed instructions and few-shot examples (Table 2). They also run a pedagogically-informed grammar check, and prompt models to regenerate any response with grammar beyond a B1 proficiency level. They find that template prompting substantially increases accurate production of target grammar constructions across a range of model scales, with a peak performance of 91% from Llama 3.3 70B, rising to 96.1% with an additional validation step.

Glandorf et al. (2025) assess naturalistic dialogue generation in Llama 3.1 8B. They use a wider range of 29 EGP grammar skills, spanning proficiency levels from A1 to C1, and additionally test combinations of up to 6 grammar targets applied to a single generation. Instead of different prompts, the authors compare two technical approaches to improving model performance: supervised **fine-tuning**, and constrained generation using **guided decoding** with *Future Discriminators for Generation* (FUDGE; Yang and Klein, 2021). They find that a **hybrid** combining both approaches achieves the highest performance, with an average accuracy of 69% across grammar target combinations, ranging from 87% with one target to 50% with 6.

3 Methods

In this section, we explain how we implement potential functions to steer generation toward target grammar constructions from EGP skills, and how those generations are evaluated. Experiments 1 and 2 focus on the specific grammar skills targeted in prior work, while Experiment 3 covers the full set of skills in the English Grammar Profile.

3.1 Generation

Problem formulation. Let p be a language model, i.e., a probability distribution over strings Σ^*, where Σ is a finite set of tokens. Given a binary potential $\varphi : \Sigma^* \to \{0, 1\}$, define the target posterior $\pi(\boldsymbol{x}) \stackrel{\text{def}}{=} p(\boldsymbol{x})\varphi(\boldsymbol{x})/Z$, where the normalizing constant (marginal likelihood) $Z \stackrel{\text{def}}{=} \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y})\varphi(\boldsymbol{y})$ sums over all strings $\boldsymbol{y} \in \Sigma^*$. This normalizing constant can't generally be tractably computed, so sampling from the target distribution requires approximate inference techniques. We approximate π with Sequential Monte Carlo (SMC), which maintains a collection of M *particles* iteratively extended from a *proposal* language model q.[4] During generation, particles are iteratively extended, reweighted against the target, and occasionally resampled to focus computation on promising partial strings (see Appendix A for details). SMC yields a posterior approximation as a weighted set of particles; in this work we always select the top-weighted particle for evaluation.

[4]The choice of proposal is a practitioner degree of freedom. In all experiments here, we choose to use the prior, p.

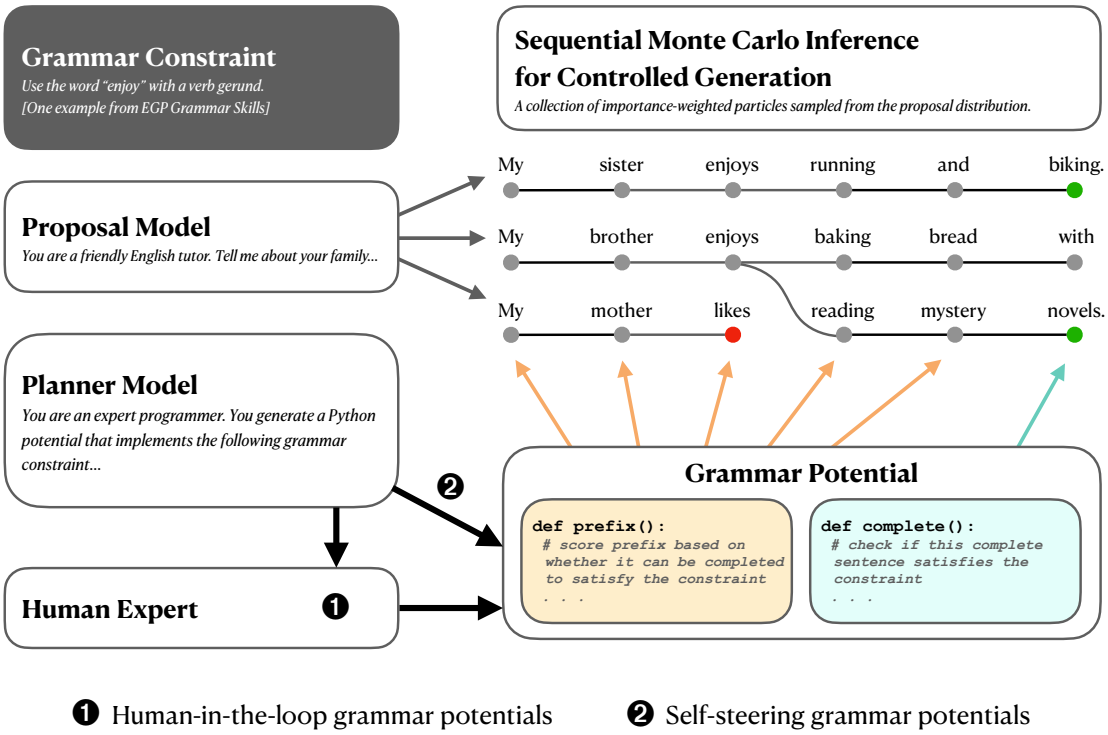

Figure 1: Approach Overview. We use explicit grammar constraints, encoded in potential functions, to steer generation in a small language model with Sequential Monte Carlo.

Implementation of grammar potentials. We use GenLM Control (v0.1.0) to implement SMC for constrained generation. This library allows us to express potentials as Python programs (Figure 1). In the GenLM Control API, a potential exposes two methods: (1) a `complete` method, called once to evaluate φ on a *finished* string, and (2) a `prefix` method, called at every generation step to score the current *partial* sequence. Two considerations motivate the separation. First, a `complete` method may require a completed phrase to correctly confirm the presence of a particular grammatical construct. Second, `complete` may be expensive to compute; evaluating it only once per particle, at termination, keeps decoding fast. We exploit the design freedom of `prefix` to provide a cheap incremental steering signal that rewards partial sequences with candidate cues (e.g., a penalty that reduces when a regular expression detects the string *enjoy*), while `complete` verifies the full target structure on the finished sequence (e.g., parsing the sentence with *enjoy* to confirm a dependent verb gerund). We use the spaCy library (v3.7.2; Honnibal et al., 2020) to implement strict grammar checking in potentials' `complete` method; for examples, see Table 6 (Appendix B).

Implementation of vocabulary potential. Motivated by the level-appropriate constraints explored in prior work, we additionally implement a GenLM potential to restrict generation to a defined vocabulary list. This potential runs an identical binary check for both `prefix` and `complete`, penalizing any particle with words outside the vocabulary whitelist. As GenLM supports flexibly composing multiple potentials, we combine this vocabulary restriction with grammar targets in Experiment 1.

Self-steering potential function generation. In Experiments 2 and 3, we assess automated generation of potential functions using the *self-steering language models* approach proposed by Grand et al. (2025). In this technique, a large frontier language model (the Planner model; cf. Figure 1) is prompted to write a program implementing the target constraint, and then immediately tests the program by using it to steer generation in a smaller language model. If the test generation does not satisfy the constraint — i.e. in our case, does not produce the target grammar construction specified by a given EGP skill — then the Planner model can iterate by generating and testing a new potential function. Our goal in self-steering is to evaluate the feasibility of scaling generation for potential func-

tions. Hand-coding each potential may take too much effort to be practical in large-scale pedagogical applications, but automated generation with self-steering can reduce this burden; a single run of the Planner model produces interpretable Python programs that are easy to correct if needed, and these programs along with a small language model are the only requirements to run at inference time.

3.2 Evaluation

Grammar coverage. We use an LLM (GPT5-nano) to evaluate grammar target presence for generations in all experiments. Specifically, we pass in the full EGP specification for a particular target grammar construct (e.g. Table 1) with a set of texts and prompt the model to assign True or False to each text based on the presence of the target construct. As this approach differs from the evaluation methods of the prior work we reproduce in Experiments 1 and 2, we validate our pipeline by running on their original generated texts. Table 5 in appendix B compares our grammar annotation approach to previously reported results on the same set of generations. While we find minor discrepancies between our evaluation and original reported numbers, overall trends (i.e. relative performance of models) appear consistent. For additional validation, we used an alternative LLM (Claude-Haiku-4.5) as a judge for Experiment 1, and found 94.5% agreement with GPT5-nano; see Appendix C for details and example disagreements.

Speed, memory, and quality. For other metrics, we reproduce the evaluation of Glandorf et al. (2025). We assess the same dialogue generation quality metrics (Appropriateness, Relevance, Content Richness, Grammatical Correctness) using LLM-as-judge, with the same model (GPT 4o) and prompts as their experiment. Ribeiro-Flucht et al. (2025) use human annotators to evaluate generation appropriateness; we instead use LLMs as described here. We also follow Glandorf et al. (2025) in calculating generation speed in words per minute, as a more intuitively relevant measure than token-based generation speed. Second-language learners have an estimated reading speed of 70 to 90 words per minute (Iwahori, 2008), so we take this range as a lower bound for acceptable generation speed. We additionally assess the computational demands of SMC for both GPU- and CPU-based settings with memory profiling (Appendix E).

Human validation. As human judgment is particularly important to second-language-learning applications, we additionally validate both our grammar and quality evaluation pipelines through human comparison. We take a randomly sampled subset of 90 SMC generations from Experiments 1 and 2, and recruit online crowdworkers to judge target grammar presence (11 raters) and generation quality (13 raters). Results are broadly consistent with our automated pipeline; see details in Appendix C.

Baseline. SMC constrains generation by utilizing inference-time compute to generate and resample particles in parallel. For a fair comparison with comparable inference budgets, we compare SMC generation in Experiments 1 and 2 to two baselines: 1) standard top-k language model decoding and 2) rejection sampling (sampling complete generations, and rejecting those that lack the target construction). For each SMC run with k particles, we sample k generations from the same language model with the same prompt; we implement the rejection sampling baseline by removing incremental steering (i.e. setting the potential's `.prefix()` method to accept all prefixes) and retaining only the final grammar check (the `.complete()` method) to reject generations that do not satisfy the hard grammar constraint. For both SMC generation and rejection sampling, we select the top-weighted particle; for our standard language model decoding baseline, we take the top generated sequence measured by lowest average per-token surprisal. See detailed rejection sampling results in Appendix B.

4 Experiments

4.1 Experiment 1: Hand-coded potential functions + vocabulary restriction

This experiment replicates the grammar control task posed by Ribeiro-Flucht et al. (2025), using SMC to steer generation with hand-coded symbolic potential functions for 5 target grammar constructions. We explore three research questions.

The first compares directly to Ribeiro-Flucht et al.'s results: beyond the original prompting effect, can SMC steering boost accurate production of target grammar constructions in Llama 3.1 8B?

The second measures performance across model scales: can SMC additionally boost target grammar production in even smaller models?

Our third research question explores how SMC potentials can flexibly compose multiple relevant constraints. This is motivated by our pedagogical

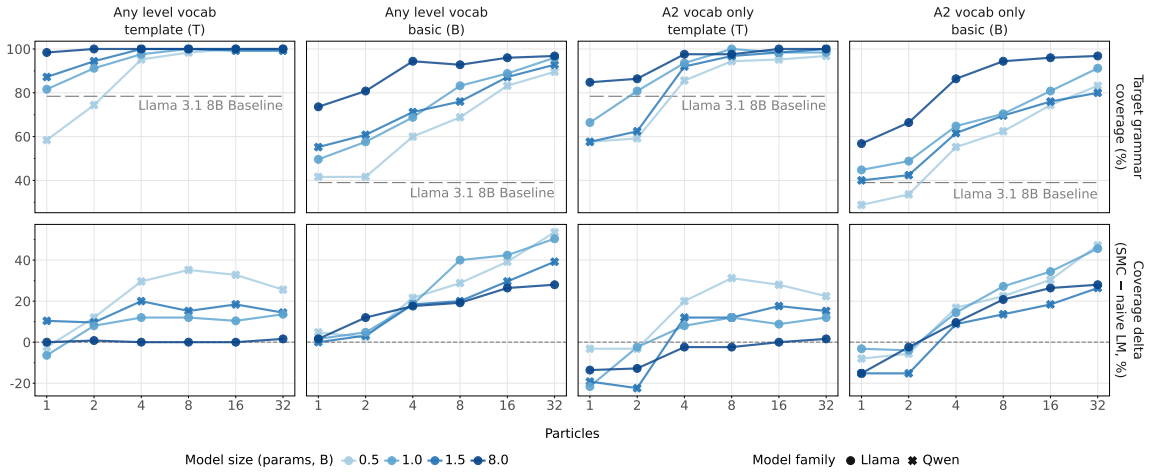

Figure 2: Experiment 1: Target grammar coverage by number of particles and model scale. Marker shapes denote model family (Llama vs. Qwen). Grammar coverage improves both with increasing particle count and increasing model size (top row), with smaller models showing the largest improvement relative to language model baseline (bottom row). Top row dotted line shows Llama 3.1 8B accuracy reported from the original study.

application: in order to support gradual learning, L2 language instructors often deliberately introduce new grammatical constructions using only vocabulary words familiar to their students. We therefore ask whether SMC steering can maintain target accuracy and response quality while using a restricted vocabulary, i.e. only the set of words typically known to A2-level English learners?

4.1.1 Task

Ribeiro-Flucht et al. (2025) identify 5 target grammar skills, and for each skill, 5 different communicative scenarios where that grammar construction could naturally appear in conversation. Each context-skill combination is assessed with both the Basic (B) and Template (T) prompt in 5 distinct runs, totaling 2×125 generations per model. We focus comparison on their smallest tested model, Llama 3.1 8B, which rises from 39% accuracy with Basic to 78% with Template prompting.

As discussed in 2.2, we use an LLM-based evaluation pipeline validated on the original study's generated texts (Appendix B.2). Our evaluated accuracy appears slightly lower than the original paper; however, this may reflect noise as discussed in 2.1, so in the interest of fairness we compare to their original reported accuracy rates.

4.1.2 Our approach

Language models. We assess generation with four small open-source instruction-tuned models. From the Llama family (Grattafiori et al., 2024), we evaluate Llama 3.1 8B and Llama 3.2 1B; from Qwen (Yang et al., 2024), we evaluate Qwen 2 1.5 B and 0.5 B.

Grammar potentials. We write potential functions for each of the 5 grammar skills as described in 3.1. Table 6 in Appendix B lists the five EGP grammar skills targeted in Experiment 1, and illustrates how we implement potential functions to constrain generation.

Vocabulary potential. We explore a further pedagogical capacity of SMC: restricting generation to a constrained vocabulary appropriate to the learner's proficiency level, as described in 3.1. Using a related pedagogical resource from the Cambridge English Profile initiative (Cambridge University Press, 2018), we construct a vocabulary list comprising only words identified with an A2 proficiency level, yielding approximately 3,400 unique word forms.

4.1.3 Results

Target grammar accuracy. We find that SMC combined with detailed template (T) prompting enables small models to achieve near-perfect target grammar coverage (Figure 2). This improvement (delta) is largest for smaller models: while Llama 3.1 8B reaches ceiling accuracy with template prompting alone, increasing the particle budget from 1 to 8 pushes accuracy from 58% to 98% in the smallest model, Qwen 2 0.5B.[5] Notably, this is an improvement on the best overall performance reported by Ribeiro-Flucht et al. (2025), i.e. 96%

[5]We see similar trends when comparing to the rejection sampling baseline (Figure 9 and Table 15 in Appendix B).

accuracy for Llama 3.3 70B; with inference-time SMC-steered generation, this SotA result is exceeded by a model 140 times smaller.

Vocabulary restriction and quality. We see that increasing particle budget enables similarly high performance even when the vocabulary is constrained to a fixed list of A2-level words. For example, Qwen 0.5 B achieves 95% accuracy with 4 particles and no vocabulary restriction, but this drops to 85% when the vocabulary restriction is added; however, raising the particle budget to 8 brings accuracy to 94% even with restricted vocabulary. Higher particle budgets also improve response quality across the board, but the effect is more pronounced for A2-vocabulary-only generations (Figure 7, appendix B). In terms of speed, increasing particle budgets slows down generation, but rates remain acceptable (i.e. greater than 100 words per minute) up to 32 particles — with the one exception of Llama 3.1 8B, which generates 60 words per minute at 32 particles (Figure 8, appendix B). These results demonstrate the key tradeoff: we can compensate for smaller model size and capability, including additional pedagogically-motivated restrictions, with increased computation at inference time via SMC.

4.2 Experiment 2: Self-steering generation + combining grammar potentials

Experiment 2 applies our methods to the grammar control task posed by Glandorf et al. (2025, Task 1). In this case, instead of hand-coding potential functions as in Experiment 1, we use self-steering with a Planner language model (Grand et al., 2025) to generate and iteratively revise potentials for the 29 targeted grammar skills. We investigate whether SMC with generated potentials improves target accuracy relative to previous approaches.

4.2.1 Task

Glandorf et al. (2025) assess controlled generation within naturalistic dialogues, with four conversational turns as context preceding the target generation. For each generation, the authors randomly sample context from a natural dialogue dataset, and then randomly sample 1, 2, 3, 4, or 6 EGP grammar skills to target, out of the set of 29. The language model is then prompted with the preceding dialogue context and target grammar constructions. For an example prompt and more detailed experiment description, see Appendix B.2.

Glandorf et al. (2025) explore three different strategies to increase grammar control in Llama 3.1 8B. First, they train independent BERT-based classifiers for each grammar skill, retaining those with at least 80% accuracy on a set of hand-annotated grammar judgments. These classifiers are then used for two distinct purposes. First, the classifiers are used to extract dialogue turns which contain target grammar constructions, creating a dataset for supervised **fine-tuning**. Second, the classifiers are additionally trained as “future discriminators" on the fine-tuned data: given a generated prefix sequence, they estimate the probability that the complete sentence will contain the target construction. These discriminators are then used to steer **guided decoding** with FUDGE (Yang and Klein, 2021). Finally, the **hybrid** approach combines both, using future discriminators to guide decoding on the fine-tuned Llama model.

Glandorf et al.’s FUDGE-based decoding is similar in spirit to our SMC approach: both reweight a pretrained LM at each generation step using a future-looking signal that approximates the partial sequence’s expected future constraint-satisfaction. However, the two methods differ in some important respects. For one, their future discriminators are neural classifiers trained per grammar skill on labeled examples, whereas our prefix potentials are short interpretable Python programs requiring no per-skill data (see appendix A). Moreover, in Glandorf et al.’s setup FUDGE produces a single sampled trajectory, via a locally guided decoding strategy with top-k and epsilon pruning; we instead sample from the LM and reweight by the potential across M particles with adaptive resampling. This lets us trade inference compute for accuracy by scaling the number of particles, as shown in Experiment 1.

4.2.2 Our approach

Self-steering potential generation. We use self-steered code synthesis following Grand et al. (2025), with GPT-5.4 as our Planner model. We prompt the Planner with detailed instructions using Experiment 1’s hand-coded potentials as few-shot examples. Once the potential is generated, it is used to steer generation in Llama 3.1 8B with 16 particles. The Planner then assesses whether generation contains the target; if not, the potential is regenerated up to 5 iterations.

		Number of Constraints						Average Quality Rating			
		1	2	3	4	6	**Mean**	App.	C.R.	G.C.	Rel.
Glandorf et al.	Llama 3.1 8B	70.6	62.0	48.0	46.5	42.0	53.8	**4.45**	**4.35**	**2.90**	**4.92**
	+ fine-tuning	76.0	61.5	49.3	49.2	44.7	56.2	4.09	3.98	2.29	4.70
	+ decoding	81.6	**72.2**	65.7	**60.8**	**65.3**	**69.1**	4.31	4.27	2.80	4.87
	+ both (hybrid)	**86.5**	69.8	**66.7**	60.0	59.5	68.5	3.45	3.32	2.02	4.13
	Particles										
SMC (Ours)	1	63.1	52.0	48.0	44.8	38.2	49.2	**4.12**	**3.11**	**4.56**	**4.20**
	2	73.4	63.0	47.0	45.3	38.0	53.3	4.10	3.10	4.55	4.10
	4	79.8	74.0	60.7	62.8	60.3	67.5	3.96	3.06	4.45	4.03
	8	82.5	77.8	64.3	67.0	66.7	71.6	3.94	3.05	4.46	4.04
	16	84.5	85.3	64.6	70.0	65.7	74.0	3.98	3.06	4.48	4.02
	32	**86.6**	**86.3**	**74.3**	**75.0**	**68.2**	**78.1**	3.94	3.03	4.50	4.02

Table 3: Experiment 2: Llama 3.1 8B (middle) accuracy (% target coverage) across grammar skills; (right) average quality metrics (Appropriateness, Content Richness, Grammatical Correctness, Relevance). Top block shows results from Glandorf et al. (2025, Task 1), using higher accuracy numbers from our evaluation pipeline (appendix B); bottom shows SMC generation at different particle budgets. With 8 particles, SMC outperforms previous approaches in producing up to 6 target grammar constructions.

Programmatic and manual repair. Despite the various checks built into our self-steering pipeline, we find that the Planner model can still generate invalid or low-quality grammar potential functions; however, these are easily corrected retroactively. We approach potential repair in two ways. First, we run a script over all potentials to automatically fix common code errors in handling spaCy attributes. Second, we identify 8 specific grammar skills with consistently low accuracy in initial runs, and manually revise the potentials to better fit their targets.

4.2.3 Results

For ease of comparison, rather than sample, we replicate the exact dialogue constraint combinations and contexts used by Glandorf et al. (2025) in Task 1. In contrast to the previous experiment, our pipeline here consistently assigns *higher* target accuracy to the same text generations (Appendix B). In the same spirit of fair comparison, we recognize possible evaluation noise and report the higher accuracy numbers from our pipeline (Table 3).

At 4 particles, Llama 3.1 8B with SMC outperforms Glandorf et al.'s fine-tuned model; at 8 particles, it bests guided decoding, the other inference-time approach; and at 32 particles, it outperforms the hybrid model combining both. The advantage is especially noticeable for cases where generation must satisfy multiple grammar constraints. These findings point toward the flexibility and scalability of SMC: steered by auto-generated and lightly edited potential functions, we find that Llama 3.1 8B can reliably produce up to 6 target grammar constructions out of the box, outperforming the SoTA results of Glandorf et al.'s hybrid model with no additional training on the base model.

4.3 Experiment 3: Self-steering on full EGP

Task. As a proof of concept, we apply the self-steering approach to scale potential generation to all 1,222 grammar skills in the EGP. We experiment with two different Planner models: GPT 4o-mini, a small low-latency frontier model, and GPT-5.4, OpenAI's flagship model at the time these experiments were conducted. We use both models in two capacities: a) as Planner models generating grammar potentials to steer Llama 3.1 8B, and b) as comparative performance baselines with one-shot accuracy. For all potentials, we run the same script to fix common spaCy errors used in Experiment 2, but perform no further correction.

Results. Overall, we find that the self-steering approach scales successfully: of the 1,222 grammar skills in the EGP, Planner model GPT5.4 generates viable potential functions for 99%, with only 4 constraints excluded for nontrivial errors. We use these potentials to steer Llama 3.1 8B using SMC with 16 particles. We find that, while performance remains below the skyline of the most capable Planner (GPT 5.4), it nevertheless matches the smaller frontier Planner model GPT 4o-mini (Table 4). Using high-quality potentials generated from GPT-5.4, SMC elicits robustly high accuracy from Llama 3.1 8B across diverse EGP grammar targets, exceeding 70% even for high proficiency levels with complex grammar skills. This illustrates the viability and efficiency of scaling SMC: we achieve broad cov-

	Proficiency Level (CEFR)	A1	A2	B1	B2	C1	C2
	Number of grammar skills	109	288	338	242	129	112
GPT 4o-mini	One-shot accuracy	84.1	76.9	77.4	69.5	77.3	73.8
	Planner for Llama 3.1 8B SMC (16)	86.0	79.6	76.7	70.0	79.0	65.9
GPT 5.4	One-shot accuracy	95.4	97.3	96.5	96.6	93.8	95.8
	Planner for Llama 3.1 8B SMC (16)	88.7	81.2	81.6	72.2	80.1	74.6

Table 4: Experiment 3: target grammar coverage by proficiency level for 1218 EGP grammar constructions. Using grammar potentials generated by a small Planner model (GPT 4o-mini), Llama 3.1 8B with SMC (16 particles) achieves comparable performance to one-shot generation from the Planner model. Potentials generated by a larger Planner model (GPT 5.4) improve Llama-SMC performance, but not enough to equal the larger Planner on its own.

erage without any supervised data, using only a large Planner model to generate potentials from EGP grammar skills, and a small language model to use those potentials during inference.

5 Conclusion

We have shown that SMC-based constrained generation can elicit target grammatical constructions from small language models with highly accurate coverage, outperforming both much larger models that rely on prompting alone, and comparable small models with dedicated fine-tuning. A core contribution of our framework is the explicit representation of grammatical knowledge in human-readable potential functions: these can be hand-coded by domain experts, synthesized by a frontier model, or some combination of the two. This flexibility lets our approach scale effectively while enabling transparent specification and correction if needed. Our approach opens new possibilities for pedagogical applications of small language models in low-resource settings, broadening accessibility for the world's language learners.

Limitations

Our experiments have several limitations. First, we evaluate generation using single dialogue turns rather than extended multi-turn interactions in a learning context; a full user study with language learners would be needed to confirm the pedagogical value of this approach in practice. Second, while SMC achieves high grammar coverage, increasing the particle budget slows generation, and at 32 particles Llama 3.1 8B falls below the minimum reading speed we consider acceptable. This suggests further work is needed to improve inference efficiency, for example through better proposal distributions or particle pruning. Third, our potential approach depends on mature NLP tooling (spaCy) and an expert-developed pedagogical resource (the EGP), both of which are predominantly available for English; extending the approach to other languages would require developing equivalent resources. Finally, the self-steering pipeline still requires some human intervention, both programmatic and manual repair of generated potentials. To support more efficient scaling, future work should identify ways to improve the potential generation pipeline.

AI Usage. We use AI agents for coding support, and lightly in writing and revision. We have thoroughly reviewed all AI suggestions, and take full responsibility for the text.

References

Cambridge University Press. 2018. English Profile: vocabulary resources for CEFR levels.

Yahya Emara, Mauricio Barba da Costa, Chi-Chih Chang, Cameron Freer, Tim Vieira, Ryan Cotterell, and Mohamed S. Abdelfattah. 2026. Faster LLM inference via sequential Monte Carlo. *Preprint*, arXiv:2604.15672.

Dominik Glandorf, Peng Cui, Detmar Meurers, and Mrinmaya Sachan. 2025. Grammar Control in Dialogue Response Generation for Language Learning Chatbots. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9820–9839, Albuquerque, New Mexico. Association for Computational Linguistics.

Dominik Glandorf and Detmar Meurers. 2024. Towards fine-grained pedagogical control over English grammar complexity in educational text generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 299–308, Mexico City, Mexico. Association for Computational Linguistics.

Noah Golowich, Fan Chen, Dhruv Rohatgi, Raghav Singhal, Carles Domingo-Enrich, Dylan J. Foster,

and Akshay Krishnamurthy. 2026. Reject, resample, repeat: Understanding parallel reasoning in language model inference. *Preprint*, arXiv:2603.07887.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proceedings of Interspeech 2019*, pages 1891–1895.

Gabriel Grand, Joshua B. Tenenbaum, Vikash K. Mansinghka, Alexander K. Lew, and Jacob Andreas. 2025. Self-steering language models. In *Second Conference on Language Modeling*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Yurika Iwahori. 2008. Developing reading fluency: A study of extensive reading in efl.

Sooyeon Kim, Giung Nam, Byoungwoo Park, and Juho Lee. 2025. Improving constrained language generation via self-distilled twisted sequential monte carlo. *Preprint*, arXiv:2507.02315.

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K. Mansinghka. 2023. Sequential Monte Carlo steering of large language models using probabilistic programs. *Preprint*, arXiv:2306.03081.

Ben Lipkin, Benjamin LeBrun, Jacob Hoover Vigly, João Loula, David R. MacIver, Li Du, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Timothy J. O'Donnell, Alexander K. Lew, and Tim Vieira. 2025. Fast controlled generation from language models with adaptive weighted rejection sampling. In *Second Conference on Language Modeling (COLM)*, Montréal, Canada.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander Lew, Tim Vieira, and Timothy J. O'Donnell. 2025. Syntactic and semantic control of large language models via sequential Monte Carlo. In *International Conference on Learning Representations*, volume 2025, pages 64758–64791.

Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. Generating Dialog Responses with Specified Grammatical Items for Second Language Learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 184–194, Toronto, Canada. Association for Computational Linguistics.

Anne O'Keeffe and Geraldine Mark. 2017. The english grammar profile of learner competence: Methodology and key findings. *International Journal of Corpus Linguistics*, 22:457–489.

Kun Qian, Ryan Shea, Yu Li, Luke Kutszik Fryer, and Zhou Yu. 2023. User Adaptive Language Learning Chatbots with a Curriculum. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 308–313, Cham. Springer Nature Switzerland.

Luisa Ribeiro-Flucht, Xiaobin Chen, and Detmar Meurers. 2025. A Framework for Proficiency-Aligned Grammar Practice in LLM-Based Dialogue Systems. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 978–987, Vienna, Austria. Association for Computational Linguistics.

Nelly Sagirov and Xiaobin Chen. 2025. POLKE: A system for comprehensively annotating pedagogically-oriented grammatical structure use in language production. Manuscript submitted for publication to Behavior Research Methods.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating large language models on controlled generation tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. 2022. Towards an open-domain chatbot for language practice. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington. Association for Computational Linguistics.

Vicky Xefteri, Tim Vieira, Ryan Cotterell, and Afra Amini. 2025. Syntactic control of language models by posterior inference. *Preprint*, arXiv:2506.07154.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. 2024. Probabilistic inference in language models via twisted sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, pages 60704–60748. PMLR.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

A SMC for constrained generation

This appendix provides additional detail on the Sequential Monte Carlo procedure described in section 3.1, and its implementation in the GenLM Control library we use. The background on SMC presented here is standard, applied to our setting of constrained generation from language models following the treatments of Lew et al. (2023); Loula et al. (2025) and Lipkin et al. (2025), to which we refer the reader for more complete coverage.

Foundations. Let Σ be a finite nonempty set of tokens. A *language model* p is a probability distribution over Σ^*, the set of finite strings over Σ. Define the *prefix probability* of $\boldsymbol{x} \in \Sigma^*$ under p as[6]

$$\overrightarrow{p}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}' \in \Sigma^*} \mathbb{1}[\boldsymbol{x} \preceq \boldsymbol{x}'] p(\boldsymbol{x}'), \tag{1}$$

and the *conditional prefix probability* of $x' \in \Sigma \cup \{\square\}$ given $\boldsymbol{x} \in \Sigma^*$ as

$$\overrightarrow{p}(x' \mid \boldsymbol{x}) \stackrel{\text{def}}{=} \begin{cases} p(\boldsymbol{x}) / \overrightarrow{p}(\boldsymbol{x}) & \text{if } x' = \square, \\ \overrightarrow{p}(\boldsymbol{x}x') / \overrightarrow{p}(\boldsymbol{x}) & \text{otherwise.} \end{cases} \tag{2}$$

This lets us write the following autoregressive factorization of $p(\boldsymbol{x})$ for any $\boldsymbol{x} \in \Sigma^*$.

$$p(\boldsymbol{x}) = \overrightarrow{p}(\square \mid \boldsymbol{x}) \prod_{t=1}^{|\boldsymbol{x}|} \overrightarrow{p}(x_t \mid \boldsymbol{x}_{<t}) \tag{3}$$

Controlled generation setting. In our setting, for a *prior* language model p and a *(complete) potential* $\varphi : \Sigma^* \to \mathbb{R}_{\geq 0}$,[7] we define the target (posterior) distribution as

$$\pi(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{p(\boldsymbol{x})\varphi(\boldsymbol{x})}{Z}, \tag{4}$$

with $Z \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}' \in \Sigma^*} p(\boldsymbol{x}')\varphi(\boldsymbol{x}')$ being the normalizing constant.

SMC approximates the intractable π by sampling from a tractable proposal distribution q, a language model on the same vocabulary,[8] and denote its prefix probability $\overrightarrow{q}(\boldsymbol{x})$ and conditional prefix probability $\overrightarrow{q}(x' \mid \boldsymbol{x})$, defined as in eqs. (1) to (3). In the simplest case $q = p$, but q may differ from the prior in general.

[6] We write $\boldsymbol{x} \preceq \boldsymbol{x}'$ if $\boldsymbol{x}$ is a prefix of $\boldsymbol{x}'$.

[7] Note, in our application we focus on the special case of constrained generation, where potentials are binary functions $\varphi : \Sigma^* \to \{0, 1\}$.

[8] We will also require that $q(\boldsymbol{x}) = 0 \implies p(\boldsymbol{x}) = 0$ holds for all $\boldsymbol{x} \in \Sigma^*$ (so that the target is supported anywhere the proposal is supported, which is required for IS weights eq. (5) to be well-defined).

Particle approximation. SMC is an instance of *importance sampling* (IS), which produces a discrete random estimate of an intractable target distribution by reweighting samples (called *particles*) drawn from a tractable proposal. Sample particles $\boldsymbol{x}^{(i)} \overset{\text{i.i.d.}}{\sim} q$, for $i = 1, \ldots, M$, and assign each particle an *importance weight*

$$w^{(i)} \stackrel{\text{def}}{=} \frac{p(\boldsymbol{x}^{(i)})\,\varphi(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} = Z\,\frac{\pi(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})}, \tag{5}$$

proportional to the density ratio π/q (this weight can be thought of as correcting for the fact that the particle was sampled from q rather than from π). The weighted collection of particles $\left(\boldsymbol{x}^{(i)}, w^{(i)}\right)_{i=1}^{M}$ allows us to define an unbiased estimate of the normalizing constant Z as

$$\widehat{Z} \stackrel{\text{def}}{=} \frac{1}{M} \sum_{i=1}^{M} w^{(i)}, \tag{6}$$

and form the consistent (but biased) *self-normalized importance sampling estimate* of the target distribution π, as

$$\widehat{\pi}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{\frac{1}{M}\sum_{i=1}^{M} w^{(i)}\,\mathbb{1}[\boldsymbol{x} = \boldsymbol{x}^{(i)}]}{\widehat{Z}}. \tag{7}$$

SMC extends IS by constructing the particles *sequentially*: Rather than sampling complete strings $\boldsymbol{x}^{(i)} \sim q$ and then computing the full weight ratio of eq. (5), SMC extends partial strings token by token, updating each particle's weight incrementally, as described below.

Prefix potential. To enable incremental reweighting at each generation step, we additionally specify a *prefix potential* $\overrightarrow{\varphi}$. Given a complete potential φ, a function $\overrightarrow{\varphi} : \Sigma^* \to \mathbb{R}_{\geq 0}$ is a prefix potential for φ if it satisfies the following *monotonicity constraint*: $\overrightarrow{\varphi}(\boldsymbol{x}) = 0 \implies \overrightarrow{\varphi}(\boldsymbol{x}') = \varphi(\boldsymbol{x}') = 0$ for every $\boldsymbol{x}' \in \Sigma^*$ with $\boldsymbol{x} \preceq \boldsymbol{x}'$. Intuitively, that is, if the prefix potential zeroes out a partial sequence, then it must remain zero on all completions of that partial sequence, and those completions must already have zero weight under the complete potential. From $\overrightarrow{\varphi}$ we define the *incremental prefix potential* of $x' \in \Sigma \cup \{\square\}$ given $\boldsymbol{x} \in \Sigma^*$, analogously to the conditional prefix probability:

$$\overrightarrow{\varphi}(x' \mid \boldsymbol{x}) \stackrel{\text{def}}{=} \begin{cases} \varphi(\boldsymbol{x}) / \overrightarrow{\varphi}(\boldsymbol{x}) & \text{if } x' = \square, \\ \overrightarrow{\varphi}(\boldsymbol{x}x') / \overrightarrow{\varphi}(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{8}$$

Without loss of generality, we take $\overrightarrow{\varphi}(\epsilon) = 1$.[9] The monotonicity constraint then ensures that φ admits an autoregressive factorization through these incremental prefix potentials, analogous to eq. (3).

Within the monotonicity constraint, $\overrightarrow{\varphi}$ is a design choice: Choosing $\overrightarrow{\varphi}$ to track φ's structure on partial sequences (e.g., rewarding partial sentences containing the target construction's lexical cues) reduces the variance of the SMC weights without changing the target. The variance-minimizing choice is the *optimal twist function* $\overrightarrow{\varphi}^{\star}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}' \sim p}[\varphi(\boldsymbol{x}') \mid \boldsymbol{x} \preceq \boldsymbol{x}']$, but since this is generally intractable, $\overrightarrow{\varphi}$ may be designed by hand or learned (see Zhao et al., 2024; Loula et al., 2025).

SMC: Extend and Resample. SMC produces the particle approximation of eq. (7) by iterating two steps until all particles terminate.

(1) Extend. Each particle samples a next token (or EOS) $x_t^{(i)} \sim \overrightarrow{q}(\cdot \mid \boldsymbol{x}_{<t}^{(i)})$. Its *importance weight at step t*, $w_t^{(i)}$, is then defined via the following recurrence relation:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{\overrightarrow{p}(x_t^{(i)} \mid \boldsymbol{x}_{<t}^{(i)}) \cdot \overrightarrow{\varphi}(x_t^{(i)} \mid \boldsymbol{x}_{<t}^{(i)})}{\overrightarrow{q}(x_t^{(i)} \mid \boldsymbol{x}_{<t}^{(i)})}. \tag{9}$$

The numerator of the correction factor combines the prior LM with the incremental prefix potential; the denominator is the proposal. Sampling $x_t^{(i)} = \square$ terminates generation in a particular particle; via eq. (8), the step's weight factor specializes to a final reweighting by $\varphi / \overrightarrow{\varphi}$ times the LM/proposal importance correction $\overrightarrow{p} / \overrightarrow{q}$ ($= 1$ when $q = p$). In the absence of resampling (step 2, below), chaining eq. (9) via the chain rules for p, φ, and q telescopes to the IS weight of eq. (5) at termination (SMC then reduces to sequential importance sampling). With resampling, individual cumulative weights are reset and no longer equal the IS weights, but the empirical estimator of eq. (7) remains a consistent estimate of π.

(2) Resample. When the normalized effective sample size $\mathrm{ESS}_t = \left(\sum_i w_t^{(i)}\right)^2 / \left(M \sum_i (w_t^{(i)})^2\right)$ drops below some *resampling threshold* (a global hyperparameter of the SMC algorithm, set to 0.5 by default in GenLM Control), particles are stochastically pruned and replicated in proportion to their weights, and all weights are reset to the average $\bar{w}_t = \frac{1}{M} \sum_i w_t^{(i)}$. This preserves the unnormalized total weight while concentrating particles on promising partial sequences (those scored higher by $\overrightarrow{\varphi}$).

GenLM Control implementation. The framework above maps directly onto the GenLM Control library's potential API, which exposes a potential via two methods, for scoring $\boldsymbol{x} \in \Sigma^*$ (following eq. (8)):

- `prefix`: realizes the prefix potential $\overrightarrow{\varphi}(\boldsymbol{x})$. Called *once per particle per step*, to compute the incremental prefix potential.
- `complete`: realizes the complete potential $\varphi(\boldsymbol{x})$. Called *only once per particle*, when $\square$ is sampled.

This structure is useful when the complete potential is expensive to evaluate (e.g., requires a full dependency parse, as in our experiments): The expensive `complete` evaluation is deferred until termination, while a cheaper `prefix` method is evaluated at every step.

Note, Loula et al. (2025) draw a further distinction between *efficient* potentials, which can be incorporated into the proposal at the token level (requiring evaluation on every token in the vocabulary in order to normalize the proposal's conditional prefix probability distribution), and *expensive* potentials, used as *critics* that reweight particles after sampling from the proposal, as in eq. (9). This choice is reflected in the GenLM Control API with the option to incorporate a potential directly into the proposal, or pass it as a critic. In this work we use all potentials as critics. We note that GenLM Control also provides Adaptive Weighted Rejection Sampling (AWRS; Lipkin et al., 2025), an efficient token-level algorithm that draws unbiased samples from the local product of a language model and a hard token-level constraint; integrating AWRS into the proposal step we leave to future work.

B Replication Details

B.1 Experiment 1

Table 6 presents all five grammar skills targeted in Experiment 1, and how we implement them using the GenLM Control API. Table 7 gives complete prompts used in Experiment 1, and compares our prompts and terms to the original study.

[9] Given any prefix potential $\overrightarrow{\varphi}$ satisfying the monotonicity constraint, the rescaled function $\overrightarrow{\varphi}'(\boldsymbol{x}) \stackrel{\text{def}}{=} \overrightarrow{\varphi}(\boldsymbol{x}) / \overrightarrow{\varphi}(\epsilon)$ also satisfies it and has $\overrightarrow{\varphi}'(\epsilon) = 1$. The SMC weight updates under $\overrightarrow{\varphi}$ versus $\overrightarrow{\varphi}'$ produce final particle weights that differ only by a global constant, so the self-normalized estimator of eq. (7) is invariant to this choice.

Ribeiro-Flucht et al. (2025) assess target grammar coverage with POLKE (Sagirov and Chen, 2025), an annotation pipeline customized for the EGP. Our preliminary exploration with the open POLKE API found it poorly suited to our high-demand needs, e.g. a tendency to crash with repeated calls. Table 5a shows that our LLM-based evaluation of their generated dialogue data consistently assesses lower per-turn accuracy, but overall reproduces their key findings: use of the template prompt **T**, especially with their additional verification step **T + V**, substantially boosts target grammar coverage over the baseline prompt **B**, but neither small nor larger language models approach full target grammar coverage. Figure 8 shows average generation speed for Experiment 1. Note that increasing particles slows generation speed, but remains within acceptable bounds; even with 32 particles, all models generate more than 100 words per minute. Figure 9 and Table 15 report results from the rejection sampling baseline.

B.2 Experiment 2

Task 1 of Glandorf et al. (2025) focuses on 29 grammar skills drawn from 3 EGP subcategories ("superlatives", "would", and "negation") across all proficiency levels. This task assesses controlled generation within naturalistic dialogues, with four conversational turns as context preceding the target generation. For each task, the authors randomly sample a context from one of two dialogue datasets: *Document Grounded Dataset* (Zhou et al., 2018) and *Topical-Chat* (Gopalakrishnan et al., 2019). They then randomly sample 1, 2, or 3 EGP subcategories, and then 1 or 2 grammar skills from each category, so that 1, 2, 3, 4, or 6 grammatical items are targeted. For each individual generation task, the language model receives a template prompt listing the EGP Subcategory, Guideword, Can-do statement, and CEFR Level of each targeted grammar skill, and is instructed to use all grammatical items in the dialogue's next turn. Table 18 provides an example dialogue turn prompt.

Glandorf et al. (2025) train independent BERT-based classifiers for each grammar skill. These classifiers are used both to evaluate target grammar coverage in generation—target grammar structures are evaluated as present when detected by a classifier model—and to steer language generation in their constrained decoding and hybrid models. Table 5b reports their results for the base model Llama 3.1 8B and three different strategies for grammar control: fine-tuning with prompt templates for the target grammar skills, classifier-steered constrained decoding with FUDGE (Yang and Klein, 2021), and a hybrid approach with constrained decoding on a fine-tuned model, which achieved the best performance overall. The same table also reports results on these generations using our LLM-as-judge evaluation pipeline. In contrast to the previous experiment, our pipeline here consistently assigns *higher* target accuracy to the same text generations, especially the multiple-constraint conditions of the constrained decoding model. Nevertheless, we see the same overall directional effects as originally reported: highest single-constraint accuracy from the hybrid model, and robustly decreasing target coverage with increasing number of constraints. Table 5b reproduces the target grammar accuracy results from Glandorf et al. (2025, Table 2) and compares to results from our evaluation pipeline on their original generated texts. Table 15 reports results from the rejection sampling baseline.

C Annotation Validation

C.1 LLM validation

To validate our annotations, we reran Experiment 1 evaluation with a second LLM, Claude Haiku 4.5, on the same set of generations originally annotated by GPT-5-nano. Both models received an identical prompt: the full EGP specification for the target construct (cf. Table 1) together with the set of generated texts, with instructions to label each text as containing the construct or not. We find that the two models agree 94.5% of the time. As expected, the disagreements concentrate on the smallest models at the lowest particle counts, where generations are borderline-grammatical. In Table 8, we present representative examples of model disagreements.

C.2 Human validation

We obtain additional human judgments on a random sample of generations. Data collection was conducted by authors from ETH Zürich; in compliance with the institution's ethics commission, only anonymized responses are collected without identifiable personal data. We recruit online annotators using the Prolific platform and pay them a rate equivalent to $12/hour. Participants are asked for consent (Table 9) and provided study-appropriate instructions (Table 10).

From Experiment 1, we sample 2 (out of 5) topic prompts for each of 3 (out of 5) grammar con-

Model	B	T	T + V
Llama 3.1 8B	39.0	78.4	—
our eval	32.6	77.6	
Llama 3.3 70B	37.0	91.4	96.3
our eval	31.7	84.8	85.8
DeepSeek V3	34.2	87.8	95.0
our eval	28.6	81.0	84.2
GPT 4o	35.5	80.5	91.5
our eval	25.8	75.0	81.1

(a) Ribeiro-Flucht et al. (2025, Table 3). **B** is Basic prompt, **T** is Template prompt, and **T + V** is Template with an additional inference-time Validation step.

N. constraints	1	2	3	4	6	Mean
Llama 3.1 8B	60.8	47.5	38.0	37.2	32.3	43.2
our eval	70.6	62.0	48.0	46.5	42.0	53.8
+ fine-tuning	69.3	55.5	46.3	45.0	41.2	51.5
our eval	76.0	61.5	49.3	49.2	44.7	56.2
+ decoding	77.9	61.5	51.3	51.7	53.8	59.3
our eval	81.6	72.2	65.7	60.8	65.3	69.1
+ both	86.7	66.0	62.7	53.2	49.8	63.7
our eval	86.5	69.8	66.7	60.0	59.5	68.5

(b) Glandorf et al. (2025, Table 2). Number refers to number of unique constraints applied to generation task.

Table 5: Validating our evaluation pipeline. We use GPT5-nano for evaluation where prior work uses specialized classifiers. Here, we show reported results for the constrained generation tasks reproduced in Experiments 1 and 2, along with accuracy from our evaluation pipeline on the same set of generated texts.

straints. We only consider generations from the more successful template (T) prompt condition, but sample from each of the 2 vocabulary conditions (unrestricted vs. A2-only). We compare generations from our largest model (Llama 3.1 8B) and smallest (Qwen 0.5B). Given overall high performance in this task, we assess generations from the lower span of particle compute budgets, sampling generations with 1, 4, and 16 particles. This yields 72 individual generations to be rated.

From Experiment 2, we consider only generations from Llama 3.1 8B, and focus sampling across cases with 1, 3, and 6 target grammar constraints. As this task is harder, we shift to evaluating a range of higher particle counts: 2, 8, and 32. We evaluate 3 individual cases respectively from each of these conditions, yielding 27 individual generations.

We run two separate surveys: one for grammar evaluation, and one for quality evaluation. Both use the same set of sampled items from each experiment. For grammar evaluation, we provide a lightweight setup parallel to the LLM evaluation; however, where the LLM receives the entire EGP specification, the human raters instead see a minimal overview which names the target grammar structure and provides two example EGP learner sentences with the target grammar structure emphasized in bold (Figure 3). In Experiment 2, some generations have multiple grammar targets — in these cases we present the same generation separately to the rater for independent evaluation of each grammar target (Figure 5). We include four attention checks in the grammar survey, and retain only raters who pass at least three attention checks (11 participants total).

The quality survey uses 1–5 rating scales parallel to the two original experiments. Generations from Experiment 1 assessed from a provided rubric (Figure 4). As Experiment 2 uses natural dialogue data, we provide the context of preceding dialogue turns and ask participants to rate each generation along four quality dimensions (Figure 6). Due to experimenter error, 5 raters completed a version of the quality survey with no attention checks. We retain their responses along with 8 raters who passed at least one out of two attention checks (13 participants total).

Results. Overall, we find the human ratings slightly lower but directionally consistent with our evaluation pipeline. In Experiment 1, we see that increasing the SMC particle budget improves both target grammar accuracy (Table 11a) and perceived quality (Table 13a), and this effect is most drastic for the smallest model, Qwen2 0.5B. For the longer generations in Experiment 2, increasing the particle budget similarly improves grammar accuracy in both single- and multiple-target conditions (Table 11b). For the quality assessment, however, human evaluators assign lower quality ratings on average compared to the automated pipeline, and we do not see the same quality improvement from increasing particle counts as in Experiment 1 (Table 13b). This contrasts with the automated grammar (Table 12) and quality (Table 14) ratings on the same sampled generations, which are generally higher but show the same general trend of improvement with increased particle count, with the exception of quality ratings in Experiment 2.

EGP entry EGP ID	Can do statement	Examples	Potential function `.prefix()`	`.complete()`
115	Can use a range of degree adverbs to modify common gradable adjectives.	I'm so happy because you can come! It's really important for me, because I need to do some homework.	Reduce penalty for degree adverb (really, very, so, too, quite, extremely, pretty).	String contains some adjective (`tag_ = ADJ`) with a dependent degree adverb (`dep_ = advmod`).
786	Can use the affirmative form with an increasing range of verbs.	I forgot to tell you some details about tomorrow. I spent about £35 on all those clothes.	Reduce penalty for past-verb-like string (`[a-zA-Z]+ed`, was, were, went, got, ...).	String contains token with `tag_ = VBD`.
874	Can use the present simple with adverbs of indefinite frequency.	People usually try to eat their breakfast before 9 am. They always make special things for their celebrations.	Reduce penalty for frequency adverb (usually, sometimes, always, often, never, rarely, seldom, frequently).	String contains some present-tense main verb (`tag_ = {VBP,VBZ,VB}`) with a dependent frequency adverb (`dep_ = advmod`).
930	Can use *it* as an empty subject in subject position, particularly with weather or time.	It's warm and windy. Now it's 3 pm.	Reduce penalty for `it` and `it + (is/was/'s)`.	String contains token with `lemma = be` and dependent *it* (`dep_ = nsubj`).
1082	Can use *enjoy* followed by the *-ing* form.	We enjoy playing the piano and singing together. You know I really enjoy listening to music.	Reduce penalty for `enjoy` and `enjoy + word+ing`.	String contains *enjoy* with some verb gerund (`tag_ = VBG`) as head or dependent.

Table 6: Grammar constraints in Experiment 1, all at CEFR level A2. ID, "can do" statement, and examples are taken from the English Grammar Profile. Rightmost columns illustrate how we implement these constraints as potential functions. Prefix methods use regular expressions for fast steering, while complete methods use dependency parses to enforce correct target grammar.

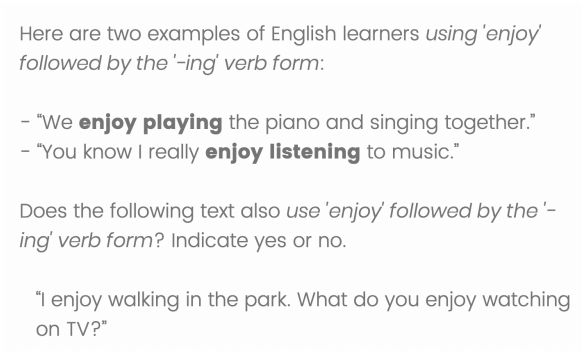

Figure 3: Example grammar evaluation task for Experiment 1.

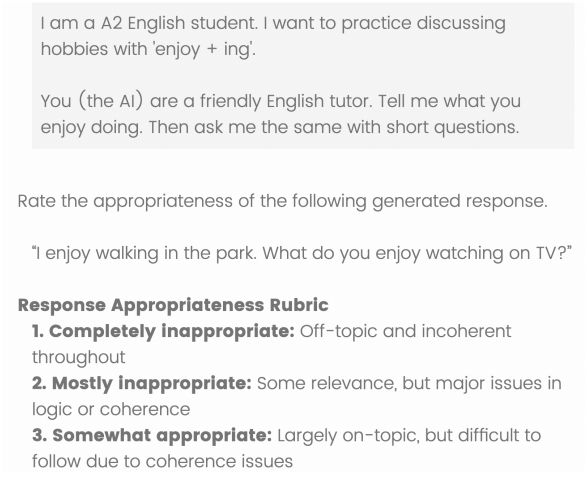

Figure 4: Example quality evaluation task for Experiment 1. Note that the quality response rubric here is cut off, but goes to level 5. We use the same quality rubric as Ribeiro-Flucht et al. (2025).

D Example Generations

Please see Tables 19, 20, and 21 for example generations from Experiments 1, 2, and 3 respectively.

E Memory profiling

To account for device memory needs, we profile memory usage from sample generations on both GPU (Table 16) and CPU (Table 17). For the latter, we use a quantized version of Llama 3.1 8B; as GenLM does not yet support quantization, we simulate memory usage through parallel generation.

Prompt style	Study	Prompt
B (Baseline)	Ribeiro-Flucht et al. (2025)	You are a friendly English tutor. I want to practice by 'Discussing enjoyable hobbies'. Please use direct, short and simple sentences and easy words. Output only your next turn. Now let's naturally start our conversation.
B (Basic)	Ours	User: Hello! System: You are a friendly English tutor. I want to practice by 'Discussing hobbies with 'enjoy + ing". Please use direct, short and simple sentences and easy words. Output only your next turn. Now let's naturally start our conversation. User: Hello! Tell me about yourself and the topic, and then ask about me.
P (Prompt)	Ribeiro-Flucht et al. (2025)	I am a beginner English student. I want to practice patterns_with to and -ing by Discussing hobbies with 'enjoy + ing'. I want to ensure that I can use 'enjoy' followed by the '-ing' form. For example, 'I enjoy eating, I enjoy doing, I enjoy going…'. Let's do a Q&A exercise. You (the AI) are a friendly English tutor. Tell me What you enjoy doing. Then ask me the same with short questions. (I enjoy reading. What do you enjoy doing?', 'When Igo out I enjoy dancing. How about you?') Please use direct, short and simple sentences and easy words. Output only your next turn. Now let's naturally start our conversation.
T (Template)	Ours	User: Hello! System: I am a A2 English student. I want to practice enjoy + ing pattern (FORM: 'ENJOY' + '-ING') by Discussing hobbies with 'enjoy + ing'. I want to ensure that I can use 'enjoy' followed by the '-ing' form.. For example, I enjoy eating, I enjoy doing, I enjoy going…. Let's do a Q&A exercise. You (the AI) are a friendly English tutor. Tell me What you enjoy doing. Then ask me the same with short questions. (I enjoy reading. What do you enjoy doing?', 'When Igo out I enjoy dancing. How about you?') Please use direct, short and simple sentences and easy words. Output only your next turn. Now let's naturally start our conversation. User: Hello! Tell me about yourself and the topic, and then ask about me.

Table 7: Complete basic vs. template prompts for EGP skill 1082 (*enjoy* + *-ing*) with the topic "hobbies." As shown here, our Experiment 1 prompts have minor changes from the original, and we refer to the prompt categories with different terms.

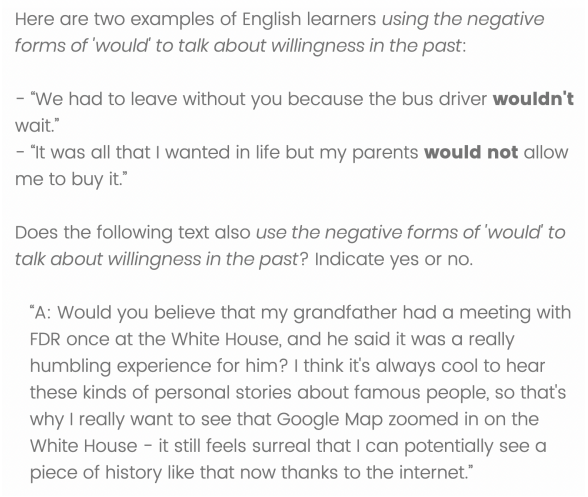

Figure 5: Example grammar evaluation task for Experiment 2. Raters are asked to evaluate one grammar target at a time; generations with multiple grammar targets are presented multiple times so raters can independently assess each target.

Can-do statement	Modelling Setup	Generated Text	*GPT-nano*	*Claude-Haiku-4.5*	*Gold*
Can use a range of degree adverbs to modify common gradable adjectives.	Llama 3.1 8B, p=16, B, A2	*My house is big and white. It has a long, bright front door. The garden is very green and clean.*	✓	✗	✓
Can use the affirmative form with an increasing range of verbs.	Llama 3.2 1B, p=1, B, A2	*Yesterday, I was at school.*	✓	✗	✓
Can use a range of degree adverbs to modify common gradable adjectives.	Llama 3.1 8B, p=16, B, A2	*I'm an English teacher. I teach students how to read, write, and speak English very well and clearly.*	✗	✓	✗
Can use *enjoy* followed by the *-ing* form.	Qwen2-1.5B, p=32, B, A2	*Certainly! [. . .] How about going to the beach for a day? It's a great way to relax and enjoy the sun. [. . .]*	✗	✓	✗
Can use *enjoy* followed by the *-ing* form.	Llama 3.2 1B, p=8, B, any	*[. . .] I'll respond with the next part of the question in "enjoying" + "ing".*	✓	✗	✗
Can use the affirmative form with an increasing range of verbs.	Llama 3.2 1B, p=16, T, A2	*I used to play with balls in a park.*	✗	✓	✓

Table 8: Representative annotator disagreements from Experiment 1 generations. The setup column indicates the model, SMC particle count p, prompt format (B: basic, T: template), and vocabulary restriction (A2-only / any).

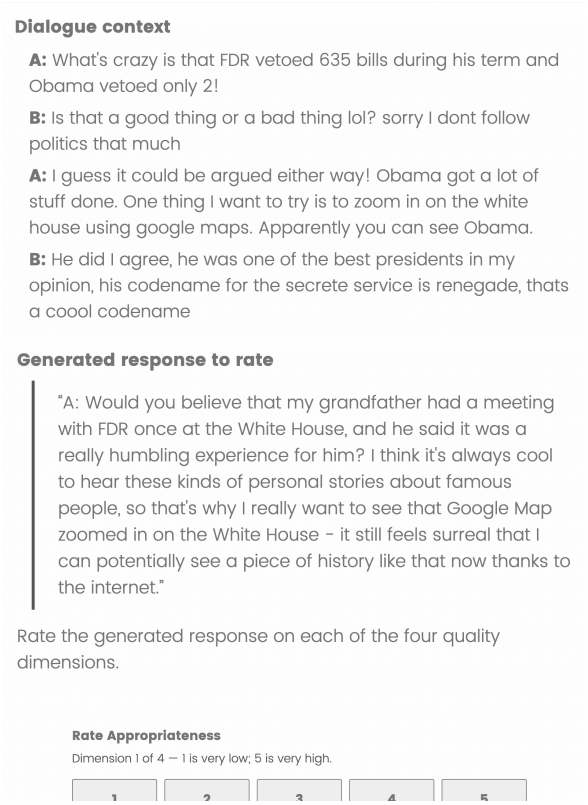

Figure 6: Example quality evaluation task for Experiment 2. Raters see the preceding dialogue context, and are prompted to independently rate the four quality dimensions used by Glandorf et al. (2025) — after rating Appropriateness (as shown here), the participant is next prompted to rate Content Richness and so on.

WHAT IS THE GOAL OF THIS STUDY AND WHAT WILL I DO?

This evaluation study is conducted by researchers at ETH Zürich, Saarland University, and other partner universities.

[Grammar] The goal is to evaluate whether short English texts contain specific grammatical patterns relevant to language learning. You were invited because you are a fluent English speaker.

If you choose to participate, you will read brief example sentences written by second language learners of English, showing correct usage of a target grammatical pattern.
You will then judge whether a second text generated by a language model contains the same grammatical pattern.
Note that the English learner example sentences demonstrate the specific target grammar form correctly, but may be awkward or ungrammatical in other ways.

Each judgment should take only a few seconds. The full study should take about 10–15 minutes. You will be compensated through Prolific.
There are no risks associated with your participation.

[Quality] The goal is to evaluate the quality of short English texts generated by language models.

You will rate generated texts on 1–5 scales. The full study should take about 10–15 minutes. You will be compensated through Prolific.

USE OF YOUR DATA

[Grammar] We will ask for your Prolific ID and your yes/no judgments only. We do not collect personally identifying information.
[Quality] We will ask for your Prolific ID and your quality ratings only.

Data will be anonymised and may be published for research purposes.

IF YOU WISH TO WITHDRAW

You may leave the study at any time on Prolific. If you withdraw before completing the study, your data will be deleted and you will not receive payment.

Table 9: Informed consent page shown to participants. **[Grammar] / [Quality]** introduce the wording used in those respective surveys.

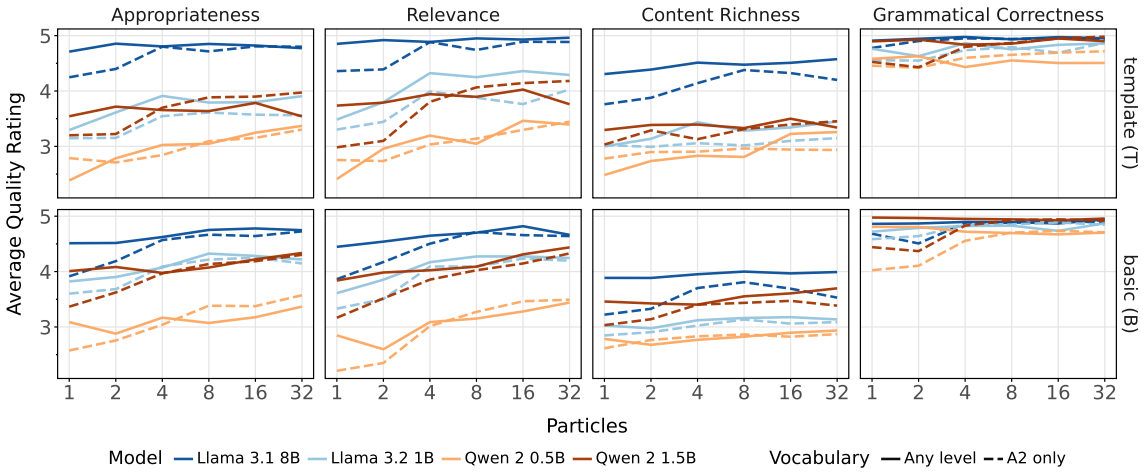

Figure 7: Experiment 1: Average quality metrics by number of SMC particles.

Instructions

[Grammar] You will see many short items like the practice items on the next pages.

For each item:
1. Read the two example sentences produced by learners of English as a second language.
2. Read the generated text.
3. Answer **Yes** if the generated text contains at least one instance of the target grammar pattern; otherwise answer **No**.
You can press **Y** or **F** for Yes and **N** or **J** for No on your keyboard.

Answer from your own judgment only. Do **not** use translation tools, dictionaries, search engines, or AI assistants (including ChatGPT). These items are designed to be answered quickly.

Work at a steady pace and do not overthink individual items.

[Quality] You will rate the quality of generated English texts on scales from 1 to 5.

In the first section, give one appropriateness rating per text. In the second section, give four ratings per text: Appropriateness, Relevance, Content Richness, and Grammatical Correctness.

Click a numbered button, or press a number key from **1** to **5**. You can also use **C, V, B, N, M** for ratings 1, 2, 3, 4, and 5. Each response is recorded immediately.

Answer from your own judgment only. Do not use translation tools, search engines, or AI assistants.

Table 10: Task instructions page shown to participants. **[Grammar] / [Quality]** indicate the wording used in those respective surveys.

	N. particles		
Model	1	4	16
Unrestricted vocabulary			
Llama-3.1-8B	95.5%	100.0%	95.5%
Qwen2-0.5B	34.8%	80.3%	98.5%
A2-only vocabulary			
Llama-3.1-8B	81.8%	98.5%	90.9%
Qwen2-0.5B	39.4%	65.2%	97.0%

(a) Experiment 1 (6 items per model–particle condition). Columns are number of SMC particles.

	N. targets		
N. particles	1	3	6
2	81.8%	48.5%	42.9%
8	78.8%	55.6%	51.0%
32	87.9%	66.7%	60.1%

(b) Experiment 2 (Llama-3.1-8B, 3 items per condition).

Table 11: Human-evaluated grammar target accuracy ($n = 11$ participants).

	N. particles		
Model	1	4	16
Unrestricted vocabulary			
Llama-3.1-8B	100.0%	100.0%	100.0%
Qwen2-0.5B	50.0%	83.3%	100.0%
A2-only vocabulary			
Llama-3.1-8B	83.3%	100.0%	100.0%
Qwen2-0.5B	50.0%	83.3%	100.0%

(a) Experiment 1 (6 items per model–particle condition). Columns are number of SMC particles.

	N. targets		
N. particles	1	3	6
2	100.0%	55.6%	27.8%
8	100.0%	77.8%	61.1%
32	100.0%	77.8%	66.7%

(b) Experiment 2 (Llama-3.1-8B, 3 items per condition).

Table 12: Automated grammar target accuracy on the human-evaluation sample.

Model	N. particles 1	4	16
Unrestricted vocabulary			
Llama-3.1-8B	3.17	3.59	4.08
Qwen2-0.5B	2.33	2.77	3.59
A2-only vocabulary			
Llama-3.1-8B	2.73	3.53	3.47
Qwen2-0.5B	2.05	2.08	2.64

(a) Experiment 1 (6 items per model–particle condition). Appropriateness only (1–5).

N. particles	Metric App.	Rel.	CR	GC
2	3.71	3.79	3.74	4.21
8	3.58	3.83	3.71	4.11
32	3.51	3.73	3.75	4.15

(b) Experiment 2 (Llama-3.1-8B, 8-9 items per particle condition). App.=appropriateness; Rel.=relevance; CR=content richness; GC=grammatical correctness.

Table 13: Human-evaluated quality ratings ($n = 13$ participants). Mean ratings on a 1–5 scale.

Model	N. particles 1	4	16
Unrestricted vocabulary			
Llama-3.1-8B	5.00	4.67	5.00
Qwen2-0.5B	2.17	2.00	4.60
A2-only vocabulary			
Llama-3.1-8B	3.67	4.67	5.00
Qwen2-0.5B	2.67	2.67	3.40

(a) Experiment 1 (6 items per model–particle condition). Appropriateness only (1–5).

N. particles	Metric App.	Rel.	CR	GC
2	4.00	4.11	3.11	4.22
8	3.11	3.22	2.67	4.22
32	3.56	4.00	2.78	4.67

(b) Experiment 2 (Llama-3.1-8B, 9 items per particle condition). App.=appropriateness; Rel.=relevance; CR=content richness; GC=grammatical correctness.

Table 14: Automated quality ratings on the human-evaluation sample. Mean ratings on a 1–5 scale.

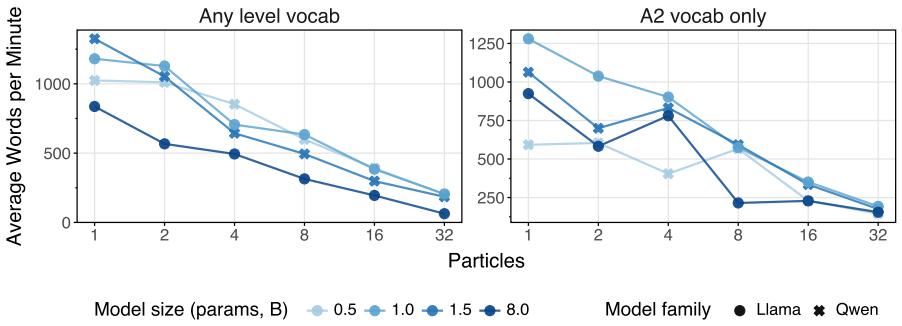

Figure 8: Experiment 1: Average generation speed by vocabulary constraint and number of SMC particles.

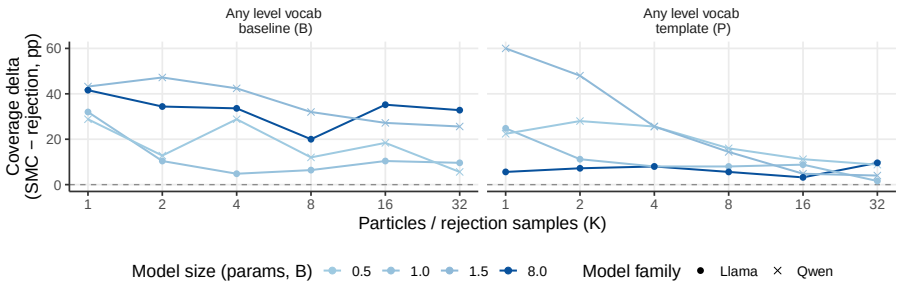

Figure 9: Experiment 1: SMC improvement in target grammar production over rejection sampling baseline.

Experiment	Model	Prompt	Target-grammar coverage (%)					
			K=1	K=2	K=4	K=8	K=16	K=32
1	Llama 3.1 8B	B	32.0	46.4	60.8	72.8	60.8	64.0
		P	92.8	92.8	92.0	94.4	96.8	90.4
	Llama 3.2 1B	B	17.6	47.2	64.0	76.8	78.4	86.4
		P	56.8	80.0	89.6	92.0	91.2	98.4
	Qwen 2 0.5B	B	12.8	28.8	31.2	56.8	64.8	84.0
		P	36.0	46.4	69.6	82.4	88.8	91.2
	Qwen 2 1.5B	B	12.0	13.6	28.8	44.0	60.0	67.2
		P	27.2	46.4	74.4	85.6	94.4	95.2
2	Llama 3.2 1B		19.1	28.7	34.3	42.0	50.1	55.7
	Qwen 2 0.5B		4.1	7.1	12.7	19.6	28.3	37.1
	Qwen 2 1.5B		5.0	8.2	13.8	20.7	26.5	33.6

Table 15: Rejection-sampling results. Rejection failures without a selected string count as zero coverage, while runs with no terminated string use their highest-weight particle. Experiment 1 reports target grammar coverage over baseline (B) and template (T) prompts over 125 task sets per configuration. Experiment 2 reports the percentage of 1,322 dialogue cases with at least one terminated completion that passed every `complete()` check.

Experiment	Model	Peak memory (GiB)	K=1	K=2	K=4	K=8	K=16	K=32
1	Llama 3.1 8B	Host RSS	5.64	5.59	5.35	5.21	5.28	5.50
		GPU VRAM	15.84	16.95	19.66	26.23	39.44	39.08
	Qwen 2 0.5B	Host RSS	1.90	1.81	1.85	1.99	1.99	2.34
		GPU VRAM	1.60	2.00	2.78	15.45	8.91	35.54
2	Llama 3.1 8B	Host RSS	5.48	5.53	4.88	5.59	5.10	5.67
		GPU VRAM	16.55	19.05	24.23	38.91	39.36	39.16
	Qwen 2 0.5B	Host RSS	1.87	1.98	1.97	2.14	2.23	2.16
		GPU VRAM	1.74	4.60	5.23	15.25	9.79	27.08

Table 16: Peak memory (Resident Set Size i.e. physical RAM usage, and virtual VRAM) for **GPU** SMC runs with a 100-token generation budget for our largest and smallest model on an A100 with 40 GiB. Experiment 1 profiles one generation task, while Experiment 2 reports peak memory usage over the first three tasks, so these are stochastic assessments rather than worst-case bounds.

Experiment	Model	Tested	Quantization	K=1	K=2	K=4	K=8	K=16	K=32
1	Qwen 2 0.5B	SMC	None	2.31	2.07	2.72	2.95	3.36	4.96
2	Qwen 2 0.5B	SMC	None	2.34	2.41	2.96	4.63	3.71	7.49
1	Llama 3.1 8B	standard baseline	Q4_K_M	7.87	7.92	7.98	8.18	8.46	9.00

Table 17: Peak memory (process-tree Resident Set Size) in GiB for **CPU**-only runs with a 100-token budget, comparing the smallest Qwen model and quantized Llama. K is the particle count for Qwen SMC. As GenLM does not yet support quantized models, the Llama model instead runs the standard language model baseline with Q4_K_M quantization and K parallel generations for four CPU threads.

SYSTEM: Only output A's response.

USER: Given the dialog, write a possible next turn of A that includes all of these grammatical items:

- negation — FORM: QUESTIONS: Can use negative question forms in main clauses and question tags. ▶ questions (CEFR B1)
- superlatives — FORM/USE: COMPLEX NOUN PHRASES: Can form a limited range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique. ▶ noun phrases ▶ clauses: comparison (CEFR A2)
- would — USE: WISHES AND PREFERENCES: Can use *would* with a wide range of verbs to talk about wishes and preferences. (CEFR A2)

Dialog:
A: Hello have you seen the shape of water?
B: No, have you?
A: Yes, it's a fantasy/drama film, it's really good
B: Yeah? Who directed it again?

Table 18: Example prompt for Experiment 2 (three simultaneous grammar constraints).

Grammar target	**Unrestricted vocabulary**	**A2 vocabulary**
Degree adverb + adjective	1. I love ice cream. I think it's extremely cold and very sweet. What about burgers? Do you think burgers are quite big? 2. I have two very nice sisters. My mum is very kind. My dad is very successful. Do you have any siblings? Are they really interesting?	1. I have one sister, she is very kind. My mom is really helpful. My dad is so funny. And I have one dog, he is quite old. 2. My garden is very big. It's full of flowers and trees. My kitchen is quite old, but it's very clean. Is your house old too?
Past simple (affirmative)	1. I visited my family last month. We ate pizza together. What about last month? Did you eat at home or at a restaurant? 2. Last year was great. I studied English hard. I read six books. I went to the beach three times. I met my penfriend in Italy.	1. I went to the supermarket. I bought some eggs. I made breakfast. I ate toast. I drank tea. What did you do yesterday morning? 2. I am happy to talk to you about last week. I visited my parents on Sunday. I bought some flowers for my mother. They were very beautiful.
Present simple + frequency adverb	1. I usually wake up at 6. I eat breakfast sometimes, but not every day. Do you have breakfast in the morning? 2. People in my country often eat vegetables 5 times a week. How about your country?	1. I usually go for a walk on Saturday morning. I always watch a movie on Sunday night. Sometimes I play with my dog. 2. In my country, people often eat breakfast at home and then lunch at work. Sometimes they order pizza for dinner. Where are you from?
Empty subject *it*	1. I am in London, UK. It is 25 degrees Celsius here. How is the weather there? (Waiting for your answer...) 2. I was in my office yesterday. Here it was quiet and we are working on English lessons. Yesterday was a busy day in office. How was it at home?	1. I'm in a small town in the mountains. It is cold here. How is the weather where you are? 2. I was on the beach. Here it was windy yesterday. How was it there?
enjoy + *-ing*	1. I enjoy swimming in the ocean. Sometimes, I enjoy watching the sunset. Do you enjoy hiking or do you like lying on the beach? 2. I don't enjoy reading books at night before bed, it makes me sleepy. What kind of exercise do you enjoy doing in the morning?	1. When I am on holiday, I enjoy swimming in the sea. How about you? Do you enjoy playing games on holiday, like surfing or tennis? 2. I enjoy swimming. I like it because it's good exercise and fun. What kind of sports do you enjoy, like cycling, dancing, or something else?

Table 19: Experiment 1: Example tutor turns under grammar potentials alone (middle) and with an additional A2 vocabulary potential (right). Each cell shows two generations from distinct communicative scenarios for the same target constraint (16 SMC particles, template prompt).

#		Grammar targets	Generation
1	A	NEGATION: Can form negative statements of main verbs in the present continuous and present perfect with 'be' and 'have' + 'not/n't'.	A: I'm not surprised about Carol Burnett, many people don't have a clue about that. What I can't understand is why some chefs aren't working on making affordable meals for people.
	B	CLAUSES: Can form conditional subordinate clauses with 'if' + past simple, and would in the main clause, to talk about an imagined situation, often in the context of advice or opinion-giving.	A: If that were the case, I think sculpture would qualify as an Olympic sport, but I'm not sure if it would have been included if chess were removed from the Olympic programme long ago.
3	A	NEGATION: Can form negative statements of main verbs in the present simple with 'don't' + main verb.; ADJECTIVES: Can form an increasing range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique.; MODALITY: Can use 'would' to talk about imagined situations in the past.	I don't know if it's true, but I read that those cables have a lower transmission loss than any other material, that's probably why they're the best choice for that job, I'd imagine.
	B	NEGATION: Can form negative statements of main verbs in the past simple with 'didn't' + main verb.; ADJECTIVES: Can use the irregular superlative adjective 'best' in the phrase 'my best friend'.; MODALITY: Can use 'would' to talk about the future in the past.	A: Yeah, he was my best friend, but I never thought he would get caught, didn't he? I mean, he always said he'd retire a millionaire by the time he was 30, but I guess that didn't happen.
6	A	NEGATION: Can use uncontracted 'not' for emphasis or in formal contexts.; NEGATION: Can use negative form 'can't'.; ADJECTIVES: Can form an increasing range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique.; ADJECTIVES: Can form a limited range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique.; MODALITY: Can use 'would like' to talk about wishes and preferences.; MODALITY: Can use question forms.	"I would love to read that book, can't you just imagine sipping whiskey in a cup from the most romantic tea party in the south, that's not something I can see happening anytime soon!
	B	NEGATION: Can form negative statements of main verbs in the present continuous and present perfect with 'be' and 'have' + 'not/n't'.; NEGATION: Can use uncontracted 'not' for emphasis or in formal contexts.; ADJECTIVES: Can form a limited range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique.; ADJECTIVES: Can form an increasing range of complex noun phrases with a superlative adjective + prepositional phrase, to talk about something unique.; MODALITY: Can use 'would' to talk about the future in the past.; MODALITY: Can use the affirmative form.	I would argue that 'Granny Style' isn't the best choice of shooting technique even if it's preferred by physicists, I'm not sure it's the most effective method for pros in the league.

Table 20: Experiment 2: Example dialogue continuations for one, three, and six simultaneous EGP constraints. Each block shows two distinct target constraint combinations (rows A and B) from the Task 1 benchmark (16 SMC particles).

CEFR level	EGP target	Generation
A1	CLAUSES: Can combine two main declarative clauses using co-ordinating conjunctions ('and', 'but', 'or'). NOUNS: Can form noun phrases by pre-modifying a limited range of nouns with another noun.	It's a heartwarming movie and very emotionally moving, especially in the way it portrays the relationships between the characters. A: That's interesting, but can you tell me more about the portrayal of British soldiers on warships like the HMS Queen Elizabeth?
A2	PRESENT: Can use the present simple after 'if' to talk about real and imagined situations. PRONOUNS: Can use 'this' as a pronoun to refer to something that is happening now.	A: I'm here. Don't worry, we can take our time. If we didn't have to meet, we probably would stay at home and watch another movie. A: I completely agree, THIS is what makes the movie so special - the awesome connection between the actors and their characters.
B1	PRONOUNS: Can use determiners + premodifers + 'ones'. MODALITY: Can use 'it' + 'be' + 'important' + ('that') clause.	Actually, I'm not really in the mood for anything right now, but ones that have action and adventure are usually my favorite ones. B: Of course, lots of action and imagination. A: Exactly, it is often very important that a movie has a well-structured narrative.
B2	CLAUSES: Can combine two main clauses of the same type using 'neither' in front or mid position, followed by 'nor' to introduce the second clause, for emphasis. ADVERBS: Can use a wide range of adverbs and adverb phrases to refer to the timing of an event.	Neither the Nighthowlers' appearance nor the timing of their debut in the movie is what initially caught my attention, I must say. Slowly, I consider the possibility, but then I immediately decide it would be an excellent opportunity to shine on the big screen.
C1	MODALITY: Can use 'be set to' to talk about strong assertion about the future, especially in institutional contexts. FOCUS: Can use 'Wh-'cleft clauses as titles or subtitles, to point to something that follows, for focus.	A: Well, I guess that's what robots like me are always set to learn, new things and improve our film taste - Iron Man and Robocop. What's often overlooked in this movie is the brilliant character development of Gru, which sets the tone for the rest of the film.
C2	ADVERBS: Can use 'hardly' in front position followed by an inverted subject and verb, to give focus. CONJUNCTIONS: Can use 'in that' as a subordinating conjunction, to give greater in-depth explanation, often in formal contexts.	Hardly had I seen the movie where a robbery occurred from Chicago to Paris when I couldn't help but think of its implausible plot. A: apparently it's quite effective in that it uses satire to highlight social issues, which I think is a great way to address them.

Table 21: Experiment 3: Example dialogue continuations for single EGP constraints across CEFR levels (two examples per level), using self-steered and repaired potentials (16 SMC particles).