

Syntactic Control of Language Models by Posterior Inference & Syntactic and Semantic Control of Large Language Models by Sequential Monte Carlo

持橋大地

統計数理研究所/国立国語研究所

daichi@ism.ac.jp

最先端NLP, 2026-8-30(日)

Syntactic Control of Language Models by Posterior Inference

Vicky Xeferi Tim Vieira Ryan Cotterell Afra Amini
{vxeferi, ryan.cotterell, afra.amini}@ethz.ch tim.f.vieira@gmail.com

ETH zürich

(ACL 2025 Findings)

SYNTACTIC AND SEMANTIC CONTROL OF LARGE LANGUAGE MODELS VIA SEQUENTIAL MONTE CARLO

João Loula^{*1} Benjamin LeBrun^{*5} Li Du^{*6} Ben Lipkin¹ Clemente Pasti² Gabriel Grand¹
Tianyu Liu² Yahya Emara² Marjorie Freedman⁸ Jason Eisner⁶ Ryan Cotterell²
Vikash Mansinghka^{‡1} Alexander K. Lew^{‡1,7} Tim Vieira^{‡2} Timothy J. O'Donnell^{‡3,4,5}
¹MIT ²ETH Zürich ³McGill ⁴Canada CIFAR AI Chair ⁵Mila ⁶Johns Hopkins ⁷Yale ⁸ISI
genlm@mit.edu

(ICLR 2025)

なぜこの論文を選んだか？

Ryan Cottrell 祭り



というのは半分冗談で...

なぜこの論文を選んだか？

- 二つとも、Sequential Monte Carlo法 (SMC, Particle Filter ともいう) をLLMのデコーディングに使った論文
- Particle FilterをNLPに使ったのは、多分私が世界最初 (NIPS 2005) → 論文では2011年以降しか触れていない
- 著者らは、NeurIPS 2023のVikashの論文に暗黙に影響を受けている様子 (VikashはCS研に滞在)

Sequential Monte Carlo Steering of Large Language Models using Probabilistic Programs

Alexander K. Lew
MIT
alexlew@mit.edu

Tan Zhi-Xuan
MIT
xuan@mit.edu

Gabriel Grand
MIT
grandg@mit.edu

Vikash K. Mansinghka
MIT
vkm@mit.edu

1本目：

Syntactic Control of Language Models by Posterior Inference

Vicky Xeferi Tim Vieira Ryan Cotterell Afra Amini
{vxeferi, ryan.cotterell, afra.amini}@ethz.ch tim.f.vieira@gmail.com

ETH zürich

(ACL 2025 Findings)

論文の扱うタスク

- 特定の文法構造を指定して、それを守るように生成するタスク
(Controlled generation)
- 技術文書、言語教育、心理実験、易しい言語の生成、言語的曖昧性の制御、..
- プログラミング言語での生成のように、基準への違反が途中で自明ではない

User:

• (S (NP (EX ?)) (VP (VBZ ?) (ADVP (RB ?)) (NP (DT ?) (NN ?))))

System:

• (S (NP (EX There)) (VP (VBZ is) (ADVP (RB always)) (NP (DT a) (NN chance))))
[$\varphi = 0.99, p = 2.95e^{-13}$]

• (S (NP (EX There)) (VP (VBZ is) (ADVP (RB never)) (NP (DT a) (NN reason))))
[$\varphi = 0.99, p = 8.66e^{-14}$]

⋮

GPT4 baseline:

• (S (NP (EX There)) (VP (VBZ is) (ADVP (RB clearly)) (NP (DT a) (~~JJ big~~) (NN problem)))) [$\varphi = 0, p = 1.90e^{-14}$]

LLMの制御

- LLMのデコーディングは、**プロンプトだけで制御できるとは限らない**
- GPT-4に、指定した構文木を持つ文を生成するように指示した結果：

Metrics	0-shot	5-shot	1-shot (gold)
correct length	42.05	43.37	65.89
exact match	17.55	19.54	31.78
structure match	20.53	25.16	44.37
F1	47.23	53.27	70.17
log φ	$-\infty$	$-\infty$	-17.41

- F値は50%程度、プロンプトの例にGoldの正解が含まれていたとしても70%程度
- どうする？

問題

- 問題を定式化すると、守るべき構文木などのタグ列 $\mathbf{t}$ が与えられたとき、テキスト $\mathbf{y}$ を確率分布

$$p(\mathbf{y}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{y})p(\mathbf{y})}{Z}$$

から生成したい

- $p(\mathbf{y})$ はLLMによる“事前確率”、 $p(\mathbf{t}|\mathbf{y})$ はタグの“尤度”、 $p(\mathbf{y}|\mathbf{t})$ は“事後分布”とみなせる → 論文のタイトル
- この分布からの生成は、自明ではない

重点サンプリング (IS)

- ランダムに $p(\mathbf{y})$ から生成して $\mathbf{t}$ を満たすかどうかを調べるのは効率が悪いので、よりよい方法は Importance sampling (重点サンプリング)
- 所望の分布 $p(x)$ に近い提案分布 $q(x)$ からのサンプリングが容易なとき、 $f(x)$ の期待値は

$$I = \int p(x) f(x) dx = \int q(x) \frac{p(x)}{q(x)} f(x) dx$$
$$\simeq \frac{1}{N} \sum_{n=1}^N \underbrace{\frac{p(x^{(n)})}{q(x^{(n)})}}_{\text{重み}} f(x^{(n)}) \quad ; \quad x^{(n)} \sim q(x)$$

モンテカルロ
サンプル

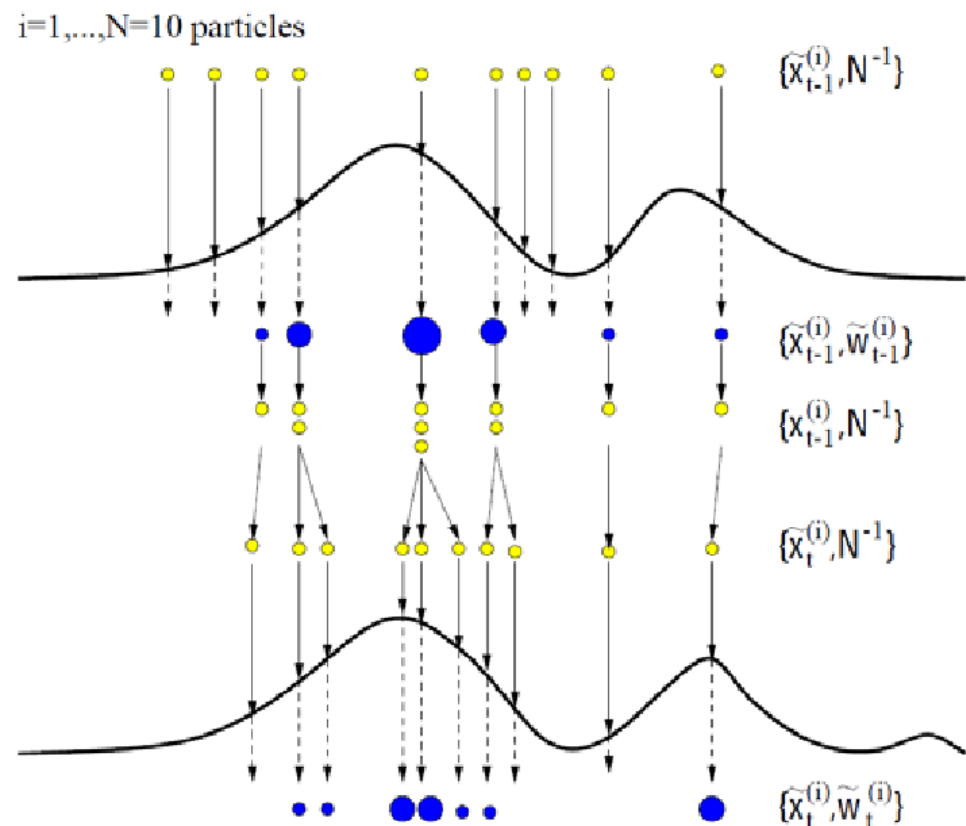
$f(x) = x$ のとき、期待値のヒストグラム近似

ただし...

- 重点サンプリングは、テキスト x を最後まで生成しないと、結果がわからない
- 実際には途中まで生成した時点で、制約をみたさないことが自明な場合が多い
→ このための方法 = Sequential Monte Carlo (SMC)

Sequential Monte Carlo法 (Particle Filter)

- 重点サンプリングを、時系列に従って行う方法
- 提案分布からのサンプリング→重みづけを繰り返す
- 重みが0に近くなると意味がないので、適宜リサンプリング
- SMC=Sequential Importance sampling (SIS) with Resampling



尤度関数と提案分布

- “Parsing as Tagging”の考え方に基づき、単語毎に2つのタグ(4種類)を振ることで構文解析を行う
Tetratagger (Kitaev+ 2020)を用いる

- 尤度関数：

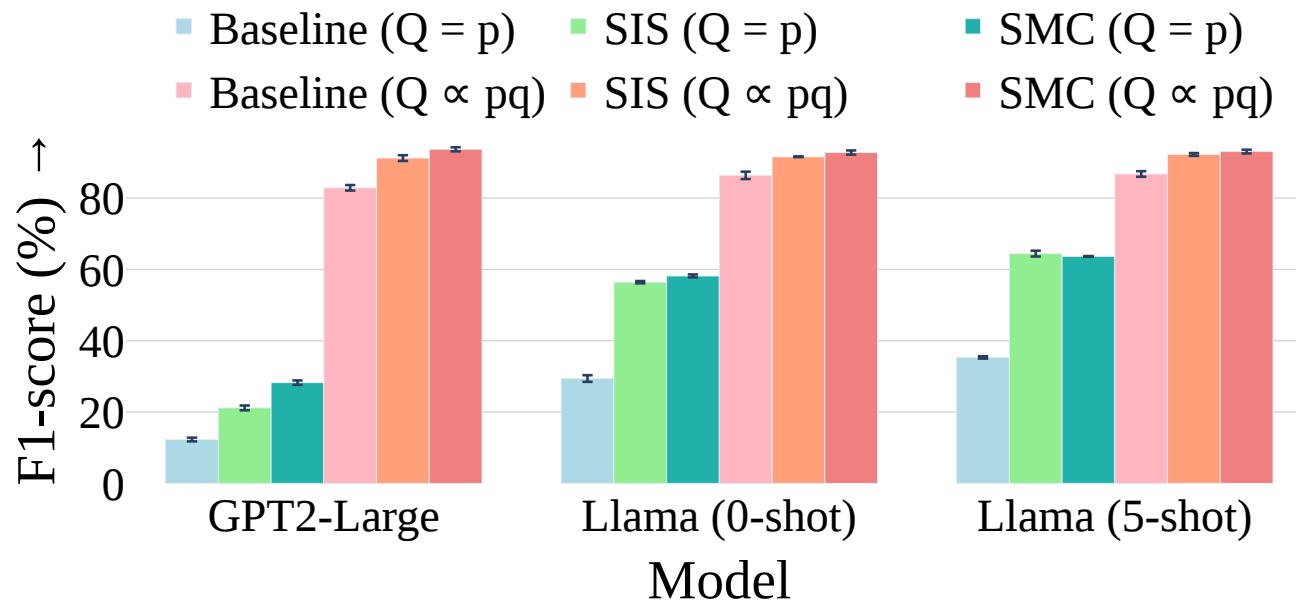
$$\psi(\mathbf{t} \mid \mathbf{y}) \stackrel{\text{def}}{=} \prod_{n=1}^N \psi(t_{2n-1} \mid \mathbf{y}_{\leq n}) \psi(t_{2n} \mid \mathbf{y}_{\leq n}), \quad (9)$$

- 提案分布：

$$Q(\mathbf{y}' \mid \mathbf{y}_{<n}) \propto p(y'_1 \mid \mathbf{y}_{<n}) q(\mathbf{y}' \mid z_n z_{n+1}), \quad (11)$$

- 次の単語 $\mathbf{y}'$ の最初のトークンの確率×タグバイグラムからの条件つき確率
- 提案分布は、それなりに緩い方がよい

実験結果



- 緑系＝提案分布が事前分布そのものの、
赤系＝提案分布が事前分布×タグ条件付き分布
- 特に前者で、SISおよびSMCでの性能向上が著しい
－ 後者の場合、性能は90%を超える

2本目：

SYNTACTIC AND SEMANTIC CONTROL OF LARGE LANGUAGE MODELS VIA SEQUENTIAL MONTE CARLO

**João Loula^{*1} Benjamin LeBrun^{*5} Li Du^{*6} Ben Lipkin¹ Clemente Pasti² Gabriel Grand¹
Tianyu Liu² Yahya Emara² Marjorie Freedman⁸ Jason Eisner⁶ Ryan Cotterell²
Vikash Mansinghka^{‡1} Alexander K. Lew^{‡1,7} Tim Vieira^{‡2} Timothy J. O'Donnell^{‡3,4,5}
¹MIT ²ETH Zürich ³McGill ⁴Canada CIFAR AI Chair ⁵Mila ⁶Johns Hopkins ⁷Yale ⁸ISI
genlm@mit.edu**

(ICLR 2025)

実験で使われているタスク

Task	Potentials		Prompt	Examples
	Φ_{eff}	Φ_{exp}		
Goal Inference	STRIPS parser	Plan simulation	Write the STRIPS goal condition for the planning problem described below [...]. The STRIPS initial condition is: [...]	<code>(:goal (and (arm-empty) (on-table b1) [...])</code>
Python Data Science	-	Error-checking with test cases	Here is a sample dataframe: [...] I'd like to add inverses of each existing column to the dataframe [...]	<code>result = df.join(df.apply(lambda x: 1/x) [...])</code>
Text-to-SQL	SQL parser	Alias and table-column checking	Here is a database schema: [...] For each stadium, how many concerts are there?	<code>SELECT T2.name, COUNT(*) FROM concert AS T1 [...]</code>
Molecular Synthesis	SMILES parser	Incremental molecule validation	Given the following list of molecules in SMILES format, write an additional molecule [...]	<code>CC1=CC2(OC=N)C(=O) [...]</code>

SMCの粒子の重みの更新

- SMCでの各粒子(=モンテカルロサンプル) $\mathbf{x}^{(i)}$ の重みの更新式
 - L_{eff} は簡単な提案分布のポテンシャル
 - Φ_{exp} は計算の重い、真のポテンシャル

$$w_t^{(i)} \stackrel{\text{def}}{=} \frac{\tilde{g}_t(\mathbf{x}_{<t}^{(i)} x_t^{(i)})}{\ell_{\text{eff}}(\mathbf{x}_{<t}^{(i)} x_t)} \quad (7a)$$

$$= \tilde{g}_0(\mathbf{x}_{<1}^{(i)}) \prod_{t'=1}^t \frac{\tilde{g}_{t'}(\mathbf{x}_{<t'}^{(i)} x_{t'}^{(i)})}{\tilde{g}_{t'-1}(\mathbf{x}_{<t'}^{(i)}) \ell_{\text{eff}}(x_{t'} | \mathbf{x}_{<t'}^{(i)})} \quad (7b)$$

$$= w_{t-1}^{(i)} \frac{\tilde{g}_t(\mathbf{x}_{<t}^{(i)} x_t^{(i)})}{\tilde{g}_{t-1}(\mathbf{x}_{<t}^{(i)}) \ell_{\text{eff}}(x_t | \mathbf{x}_{<t}^{(i)})} \quad (7c)$$

$$= w_{t-1}^{(i)} \cdot L_{\text{eff}}(\mathbf{x}_{<t}^{(i)}) \cdot \frac{\Phi_{\text{exp}}(\mathbf{x}_{<t}^{(i)} x_t^{(i)})}{\Phi_{\text{exp}}(\mathbf{x}_{<t}^{(i)})} \quad (7d)$$

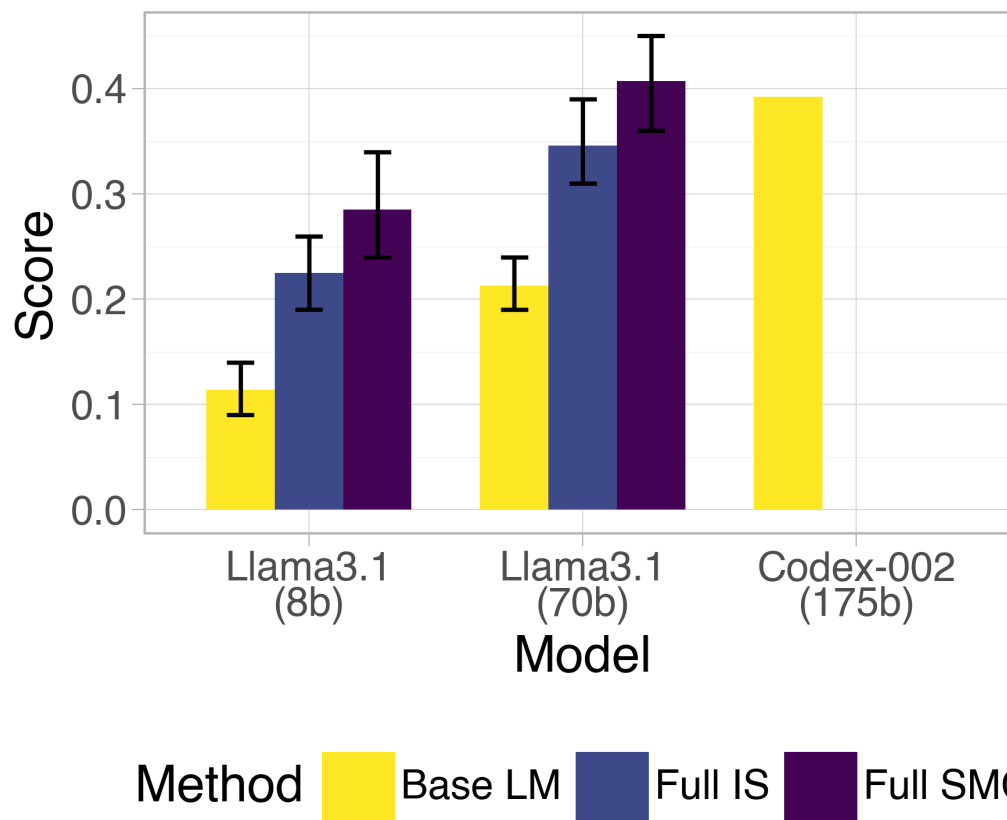
SMCのアルゴリズム

1. *Extend*. For each incomplete particle $\mathbf{x}^{(i)}$, sample $x_t^{(i)} \sim \ell_{\text{eff}}(\cdot | \mathbf{x}_{<t}^{(i)})$ and update $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} x_t^{(i)}$.
2. *Reweight*. For each extended particle $\mathbf{x}^{(i)}$, update the weight $w^{(i)} \leftarrow w^{(i)} L_{\text{eff}}(\mathbf{x}_{<t}^{(i)}) \frac{\Phi_{\text{exp}}(\mathbf{x}_{<t}^{(i)} x_t^{(i)})}{\Phi_{\text{exp}}(\mathbf{x}_{<t}^{(i)})}$.
3. *Resample*. Sample ancestor indices $a^{(1)}, \dots, a^{(N)} \stackrel{\text{i.i.d.}}{\sim} \text{Categorical}\left(\frac{w^{(1)}}{W}, \dots, \frac{w^{(N)}}{W}\right)$ where $W = \sum_{i=1}^N w^{(i)}$. Then, reassign all particles $(\mathbf{x}^{(i)}, w^{(i)}) \leftarrow (\mathbf{x}^{(a^{(i)})}, \frac{W}{N})$ for all i simultaneously.⁷

- 以上をM個の粒子 (モンテカルロサンプル)
 $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}$ について行う

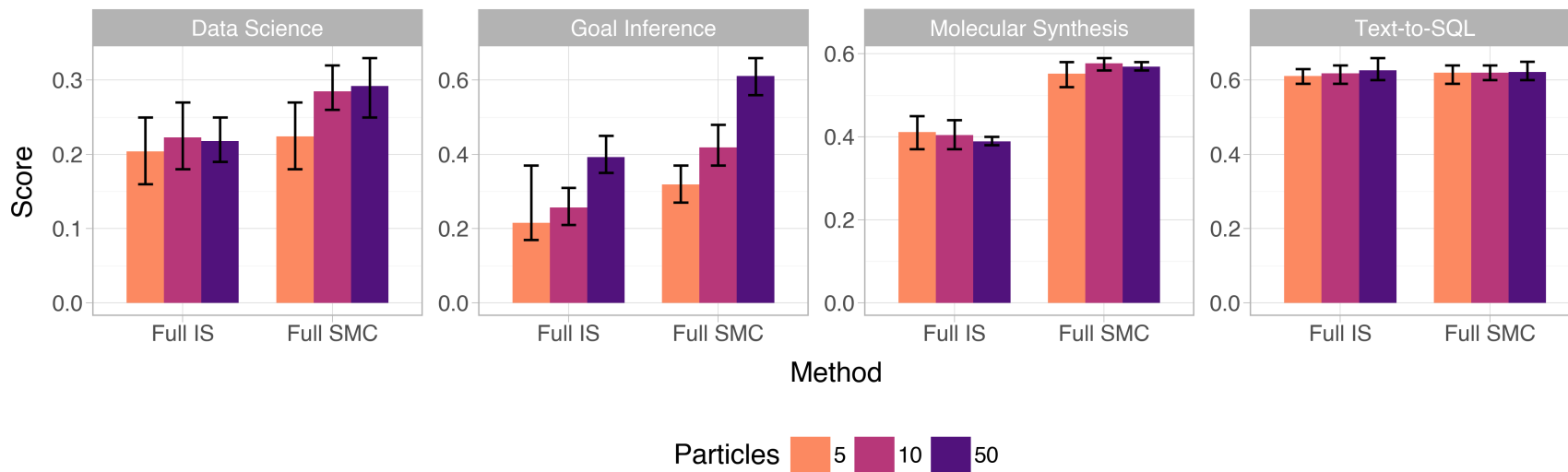
実験結果(1)

- データサイエンスのタスク(DS-1000)での結果：
小さめのLLMでも巨大・専用モデルに迫る性能



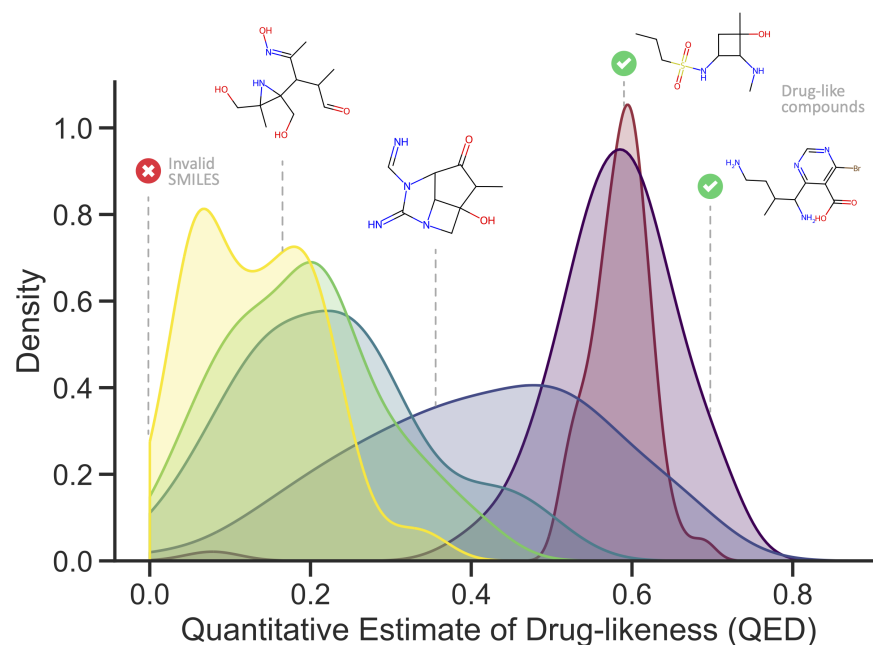
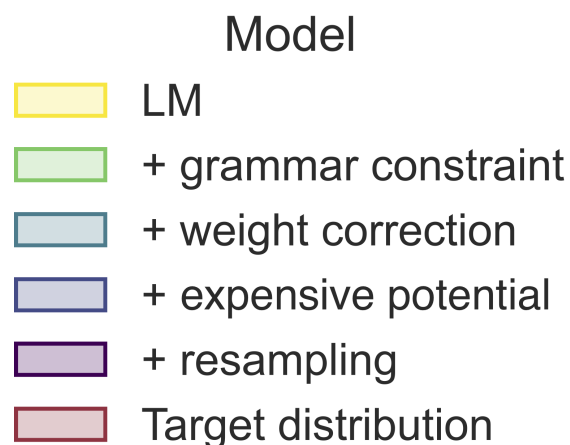
実験結果(2)

- 4種類のタスクでの性能 (色は粒子の数)



- 重点サンプリング(IS)を時系列で行うSMCの方が一貫して高性能
- 粒子の数は、タスクによって多い方がよい

真の目標分布とSMC近似分布



- SMCによる逐次モンテカルロ近似は、真の目標分布にかなり近くなっている
 - 事前分布(黄色)は、目標分布とはかなり異なる

結論

- 条件つき生成の際に、単にプロンプトを与えて上手く生成できるように「お祈り」するのではなく、確率モデルに基づいて生成を制御する
- 生成が終わってから選ぶのではなく、生成途中にスコアを計算して有望なものを優先して並列に生成
- 以上を行うための、Sequential Monte Carlo法の適用