LREC-COLING 2024

Few-shot Temporal Pruning Accelerates Diffusion Models for Text Generation

Bocheng Li[1,3], Zhujin Gao[1,3], Yongxin Zhu[2,3], Kun Yin[4], Haoyu Cao[4]
Deqiang Jiang[4], Linli Xu[1,3*]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]School of Data Science, University of Science and Technology of China
[3]State Key Laboratory of Cognitive Intelligence
[4]Tencent YouTu Lab
bcli@mail.ustc.edu.cn, GitHub @bc-li

A short summary of our work

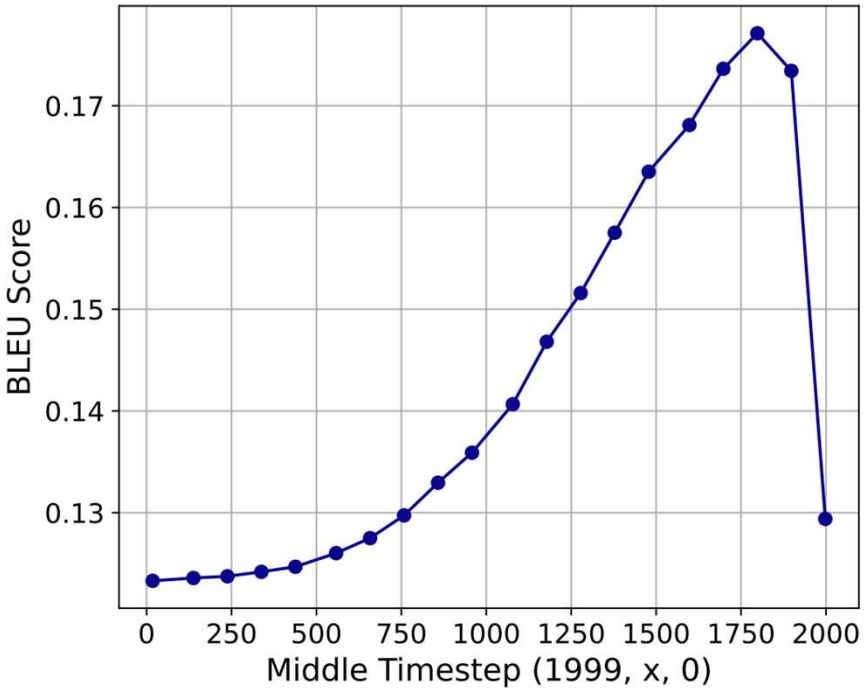
- We propose a novel method called Few-shot Temporal Pruning to accelerate Diffusion Model for faster text generation, getting a speed up for up to 400x in down to less than 1 minute of post-training optimization.

Introduction

- Previous diffusion models for text generation are slow – requiring 50-2000 sampling steps
- Most of the accelerating methods ignore the importance of the distribution of sampling steps / requires further training
- May not be applicable under tight resource and time restrictions

Observation

- 3-step sampling, paraphrasing in DiffuSeq (sampling steps can be selected from [0,1999])
- fixing the initial and final sampling steps and varying the middle step
- substantial changes in BLEU score of the generated samples!! (from 0.12 – 0.17)



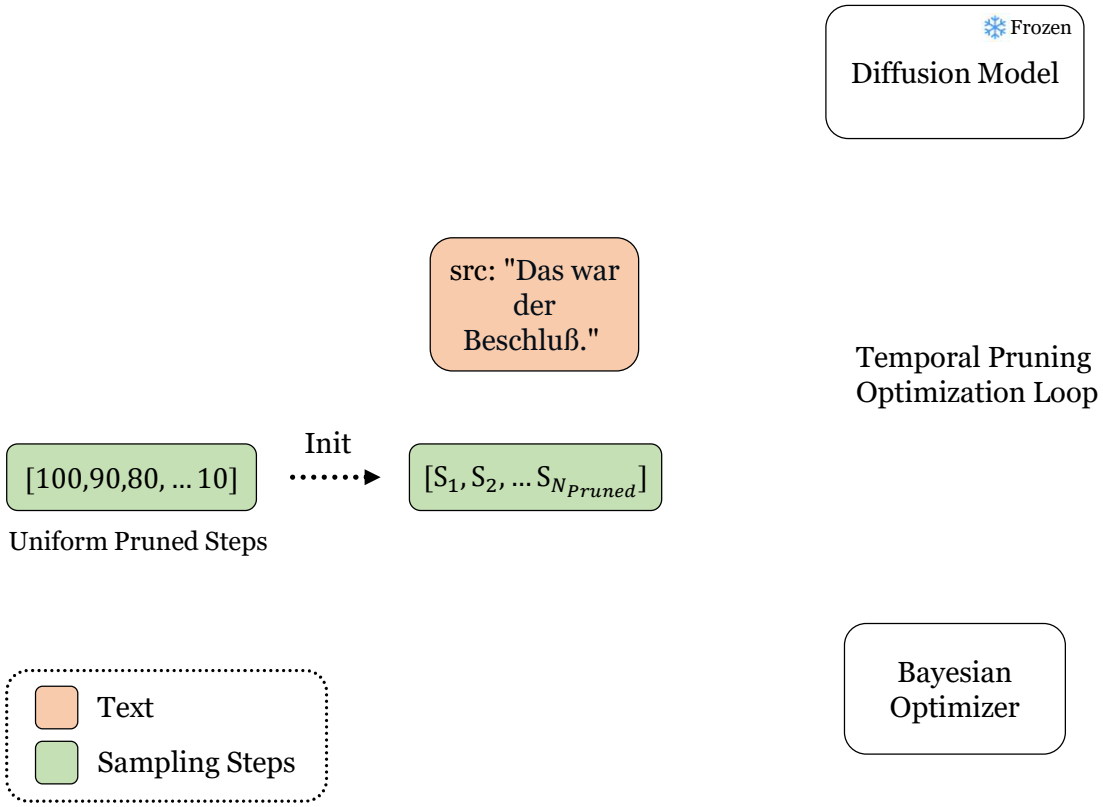
Finding the steps that really matters

Defining “significant steps” and “redundant steps”

- Significant Steps: eliminating these samping steps, the progressive refinement of samples would be disrupted, and diffusion models would no longer generate high-quality outputs.
- Redundant Steps: pruning these non-critical sampling steps enhances the efficiency of the sampling process without compromising the quality.

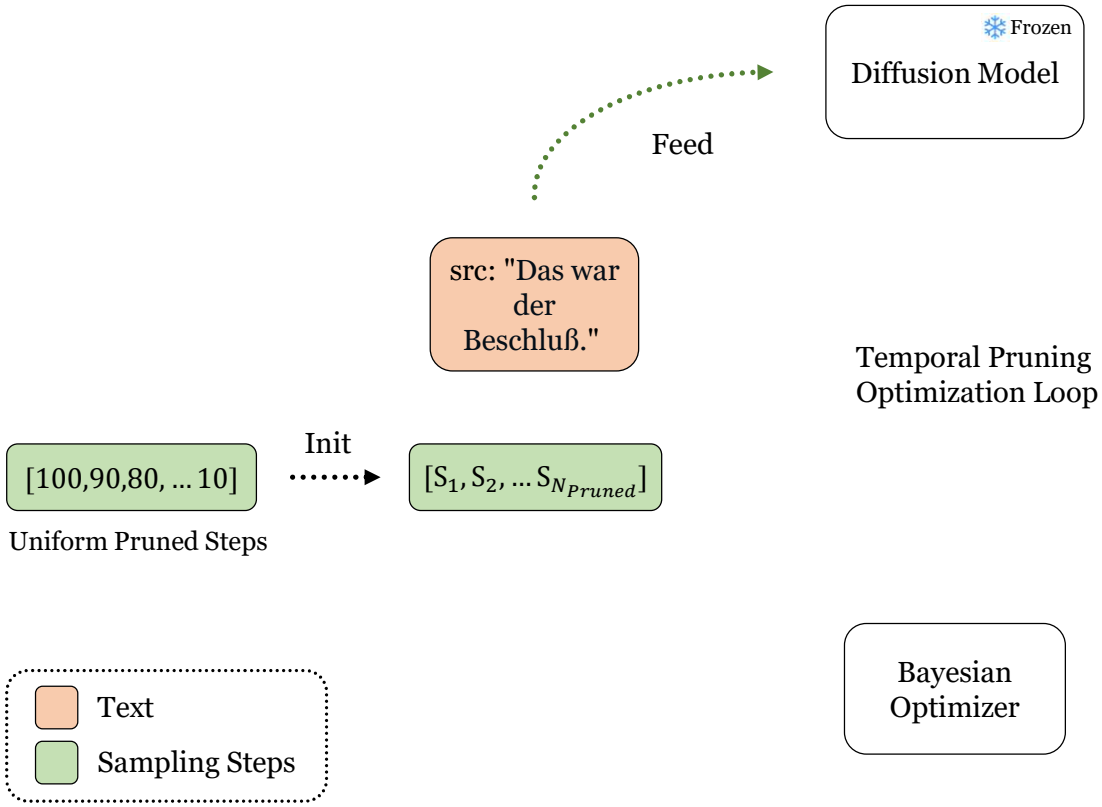
Optimization Loop of Temporal Pruning

Uniformly initialize the distribution of sampling steps $[S_1, S_2, \ldots S_{N_{Pruned}}]$



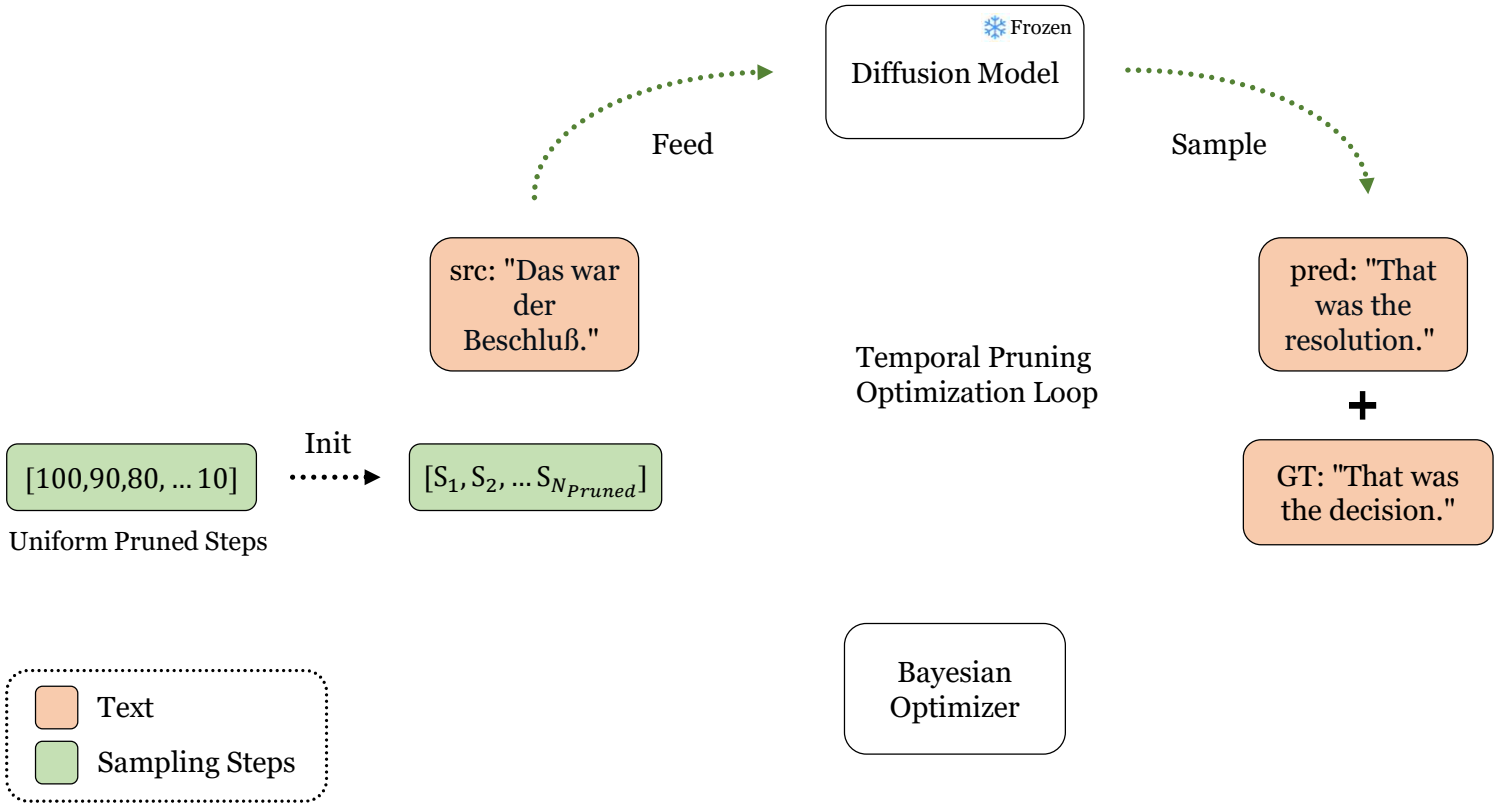
Optimization Loop of Temporal Pruning

Feed the source and distribution of sampling steps $[S_1, S_2, \ldots S_{N_{Pruned}}]$ to a frozen Diffusion model



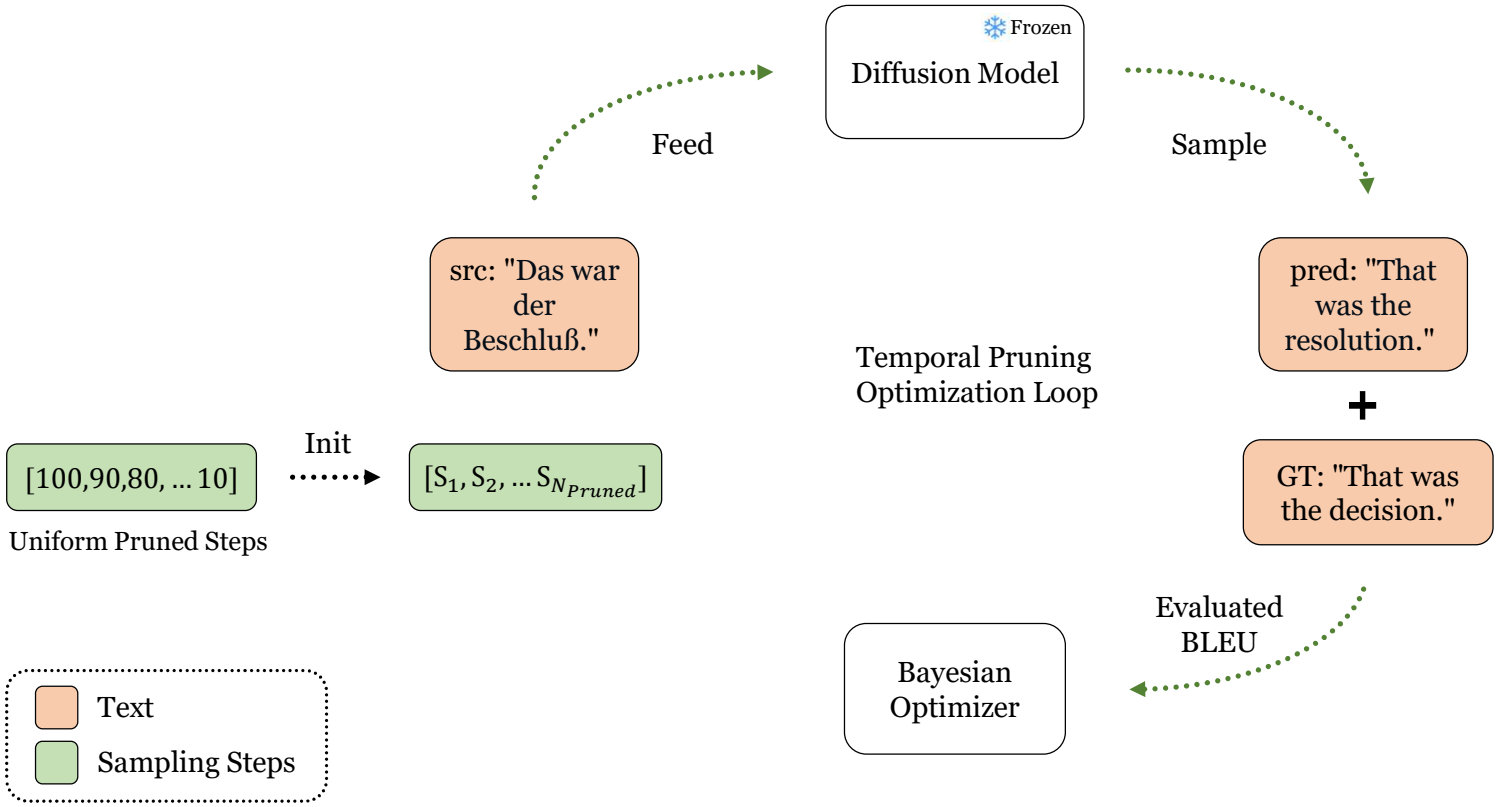
Optimization Loop of Temporal Pruning

use the source text and distribution of sampling steps $[S_1, S_2, \ldots S_{N_{Pruned}}]$ to generate new sample pred and compute BLEU with GT



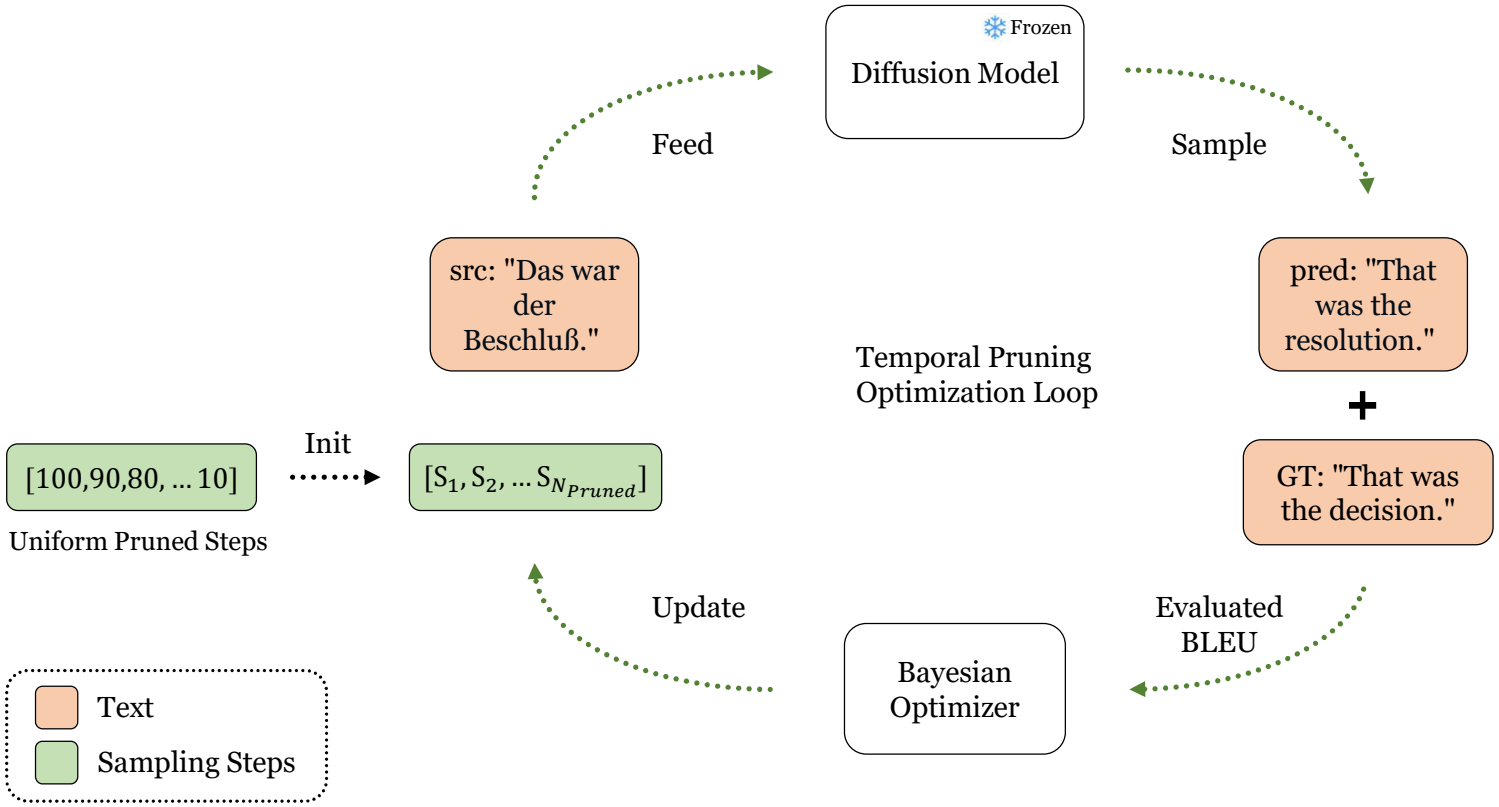
Optimization Loop of Temporal Pruning

Use a Bayesian optimizer to use evaluated BLEU and former information to generate next better sample steps



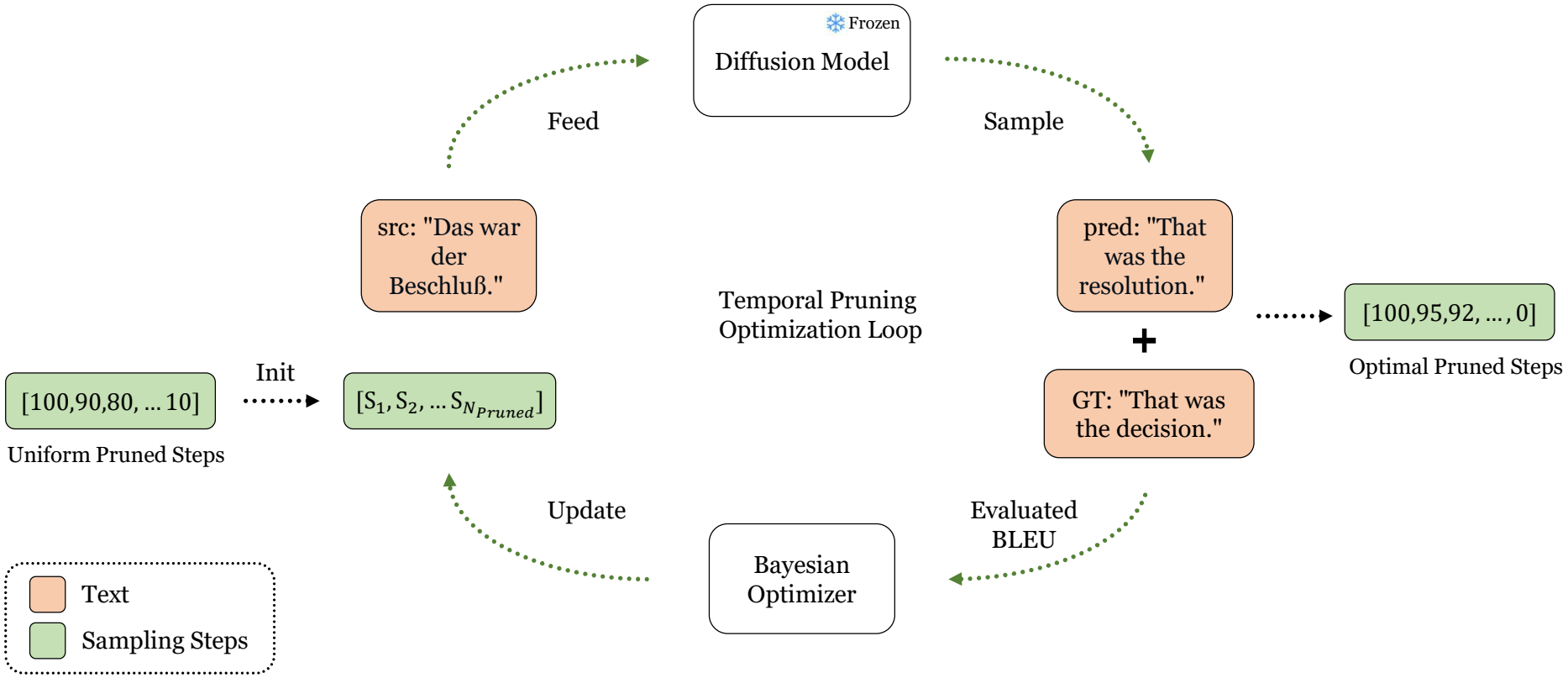
Optimization Loop of Temporal Pruning

Update the new sampling steps to $[S_1, S_2, \ldots S_{N_{Pruned}}]$



Optimization Loop of Temporal Pruning

After some optimization loops, get the optimal distribution of sampling steps



How does Bayesian optimizer optimize?

A brief introduction to Bayesian Optimization we use

- Designed for optimizing black-box, expensive functions, always used for hyper-parameter tuning, in our method, used to select next better $[S_1, S_2, \ldots S_{N_{Pruned}}]$.
- It use an observation set O to record all previous information(explored$[S_1, S_2, \ldots S_{N_{Pruned}}]$, and its corresponding R)
- Iteratively update Gaussian process posterior(it models the objective function(performance function)

$$R(\cdot) = \mathrm{BLEU}(M(\cdot), Y_{\mathrm{gt}})$$

- Calculate an inexpensive acquisition function(instead of perform a real sampling process, it is expensive!) in a subset D' of the entire set of possible distributions D.

How does Bayesian optimizer optimize?... Cont'd

A brief introduction to Bayesian Optimization we use

- D' is first initialized randomly, then explored using Limited-memory BFGS optimization algorithm(for 20 iters)
- We explore next $S = [S_1, S_2, \ldots S_{N_{Pruned}}]$ which satisfies

$$\mathbf{S}_{\text{optimal}} = \arg\max_{\mathbf{S} \in O} R(\mathbf{S})$$

Few-shot Temporal Pruning

A more data-efficient variant of Temporal Pruning

- We modify the data used to perform a whole sampling process:

src: "Das war der Beschluß."

- Initially, we use a whole validation set to search for $S_{Optimal}$.
- It is indeed slow to sample 7000+ items in one run
- Cut them down to 20 random items in the validation set
- justifies the robustness in scenarios with limited data availability

Baselines, dataset and benchmark

- Baseline:
  - Transformer-base
  - Discrete Diffusion: Multinomial Diffusion and Absorbing Diffusion
  - Continuous Diffusion: DiffuSeq
- Dataset:
  - Machine Translation: IWSLT14 DE-EN, WMT16 EN-RO, WMT14 EN-DE
  - Question Generation: Quasar-T
  - Paraphrasing: QQP dataset
- Benchmark: BLEU score(Metric), Running Time, Inference Speedup

Results for Generation Quality

Model	Steps	Shots	IWSLT14 DE-EN	WMT16 EN-RO	WMT14 EN-DE	QG	QQP
Transformer-base	-	-	34.51§	34.16§	27.53§	16.63†	27.22†
Absorbing Diffusion							
Vanilla	50	-	28.95	30.88	22.98	17.49	24.34
Vanilla	4	-	27.16	27.41	18.70	17.45	24.07
Temporal Pruning	4	Full Set	28.61	31.03	22.42	17.47	24.41
Temporal Pruning	4	20	28.12	29.51	21.69	17.47	24.21
Multinomial Diffusion							
Vanilla	50	-	13.12	4.50	0.32	17.45	24.06
Vanilla	4	-	24.23	27.80	17.19	17.08	21.52
Temporal Pruning	4	Full Set	26.96	29.69	21.44	17.48	23.70
Temporal Pruning	4	20	26.83	29.88	20.98	17.38	23.22
DiffuSeq							
Vanilla	2000	-				17.31	24.13
Vanilla	4	-	-	-	-	16.06	19.05
Temporal Pruning	4	Full Set				16.38	22.32
Temporal Pruning	4	10				16.38	21.90

Results for Running Time & Inference Speedup

Model	Shots	Total	Inference	Bayesian
DiffuSeq				
QQP	Full	19.22 min	19.22 min	0.04 s
	20	0.63 min	0.63 min	0.04 s
QG	Full	18.94 min	18.94 min	0.04 s
	20	0.66 min	0.66 min	0.04 s
Absorbing				
IWSLT14 DE-EN	Full	69.17 min	69.17 min	0.04 s
	20	0.76 min	0.76 min	0.04 s
WMT16 EN-RO	Full	22.50 min	22.50 min	0.04 s
	20	0.84 min	0.84 min	0.04 s
WMT14 EN-DE	Full	33.33 min	33.33 min	0.04 s
	20	0.84 min	0.84 min	0.05 s
QG	Full	50.00 min	50.00 min	0.04 s
	20	0.78 min	0.78 min	0.05 s
QQP	Full	45.83 min	45.83 min	0.04 s
	20	0.82 min	0.82 min	0.05 s
Multinomial				
IWSLT14 DE-EN	Full	80.00 min	80.00 min	0.04 s
	20	0.92 min	0.92 min	0.04 s
WMT16 EN-RO	Full	36.67 min	36.67 min	0.04 s
	20	0.96 min	0.96 min	0.04 s
WMT14 EN-DE	Full	51.67 min	51.67 min	0.04 s
	20	0.93 min	0.93 min	0.04 s
QG	Full	65.83 min	65.83 min	0.04 s
	20	0.90 min	0.90 min	0.04 s
QQP	Full	54.17 min	54.17 min	0.04 s
	20	0.93 min	0.93 min	0.04 s

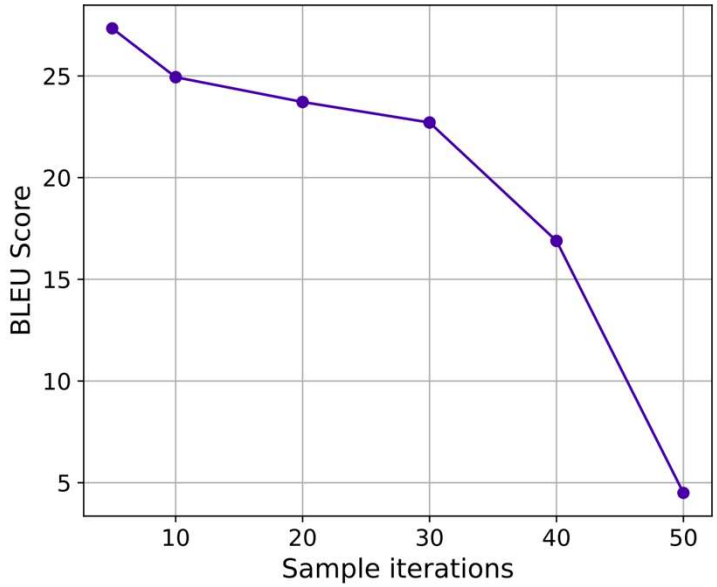
Running Time

Model	TP	Baseline	Speedup
DiffuSeq			
QQP	91.54 sps	0.226 sps	405x
QG	92.14 sps	0.218 sps	423x
Absorbing			
IWSLT14 DE-EN	182.91 sps	18.50 sps	9.88x
WMT16 EN-RO	114.98 sps	11.32 sps	10.15x
WMT14 EN-DE	115.12 sps	10.95 sps	10.51x
QG	73.40 sps	6.70 sps	10.95x
QQP	92.47 sps	9.20 sps	10.05x
Multinomial			
IWSLT14 DE-EN	134.27 sps	14.51 sps	9.25x
WMT16 EN-RO	68.86 sps	6.42 sps	10.72x
WMT14 EN-DE	64.01 sps	5.92 sps	10.81x
QG	52.17 sps	4.27 sps	12.21x
QQP	80.86 sps	6.36 sps	12.71x

Inference Speedup

Analysis: Overcoming Sampling Degradation via Temporal Pruning

- Multinomial Diffusion, after several iterations, shows a tendency of $p_\theta(x_{t-1}|x_t) \approx x_t$
  - essentially replicates the previous state
  - degradation in sampling performance
- Redundant steps negatively impact the model performance in two ways:
  - hinder sentences from reaching an optimal state
  - consume computational resources without improving the output quality after degradation



Performance degradation on the WMT16 test set using the vanilla multinomial diffusion model.

Analysis: Overcoming Sampling Degradation via Temporal Pruning

- Multinomial Diffusion, after several iterations, shows a tendency of $p_\theta(x_{t-1}|x_t) \approx x_t$
  - essentially replicates the previous state
  - degradation in sampling performance
- Redundant steps negatively impact the model performance in two ways:
  - hinder sentences from reaching an optimal state
  - consume computational resources without improving the output quality after degradation

Source: ich danke ihnen für ihre aufmerksamkeit. **Reference:** thank you for your attention.	
# Iter.	Decodes
Vanilla Multinomial Diffusion, 50 steps	
0	∘ books mindestens bridge ght dahin
10	∘ books mindestens bridge ght dahin
20	∘ books mindestens bridge ght dahin
30	∘ books mindestens bridge ght dahin
40	∘ books mindestens bridge ght dahin
50	∘ books mindestens bridge ght dahin
Multinomial Diffusion with Temporal Pruning, 4 steps	
0	∘ jähr## dadurch sprü## vege## ined depres## de## frag##
1	∘ jähr## dadurch very much for your attention .
2	∘ thank you very much for your attention .
3	∘ thank you very much for your attention .
4	∘ thank you very much for your attention .

Table 4: A comparison of samples generated from multinomial diffusion w/ and w/o Temporal Pruning on IWSLT dataset. Words are in lower case, and ## denotes the sub-word tokenization artifacts.

Analysis: Insufficient Noising at Early Steps

- A notable shift in the distribution of the optimized sampling steps
  - a tendency to concentrate at higher steps
- Higher steps are of greater importance:
  - the model is exposed to a substantial amount of noise during these stages
  - more comprehensive training

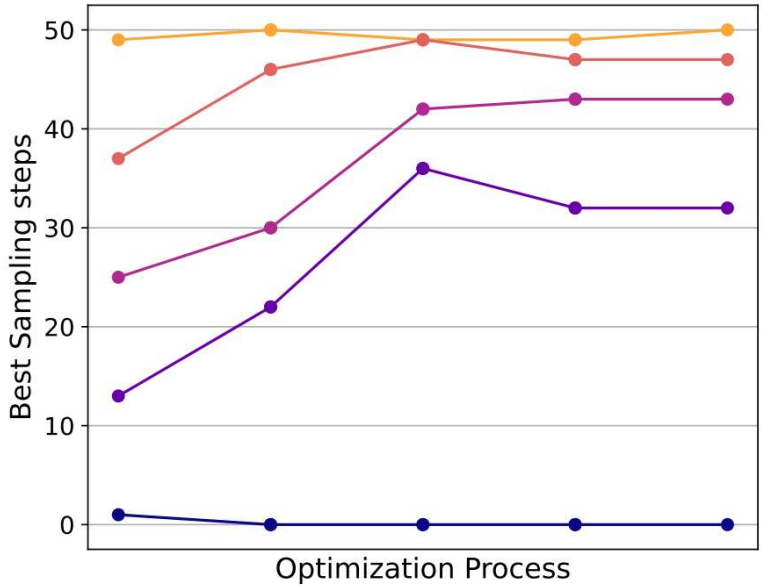


Figure 4: The distribution of sampling steps throughout the optimization process, with each color indicating an individual timestep S_i belonging to the pruned set $\mathbf{S} = [S_1, S_2, S_3, S_4, S_5]$. The colored lines from top to bottom correspond to S_1, S_2, S_3, S_4, and S_5, respectively.

Conclusion

- We present Few-shot Temporal Pruning, a robust, effective and training-free approach to accelerate diffusion models for text generation
- a thorough qualitative analysis of the effects of redundant sampling steps on model performance and the optimized distribution of sampling steps

More info

- Please keep in update with GitHub repo:
  - **https://github.com/bc-li/temporal-pruning**
- If you have other questions, feel free to contact **bcli@mail.ustc.edu.cn**

Thanks for listening!

Q&A

This research was supported by the National Natural Science Foundation of China (Grant No. 62276245), and Anhui Provincial Natural Science Foundation (Grant No. 2008085J31).