Locate&Edit: Energy-based Text Editing for Efficient, Flexible, and Faithful Controlled Text Generation

Anonymous ACL submission

Abstract

Recent approaches to controlled text generation (CTG) often involve manipulating the weights or logits of base language models (LMs) at decoding time. However, these methods are inapplicable to latest black-box LMs and ineffective at preserving the core semantics of the base LM's original generations. In this work, we propose `Locate&Edit`(`L&E`), an efficient and flexible energy-based approach to CTG, which edits text outputs from a base LM using off-the-shelf energy models. Given text outputs from the base LM, `L&E` first *locates* spans that are most relevant to constraints (e.g., toxicity) utilizing energy models, and then *edits* these spans by replacing them with more suitable alternatives. Importantly, our method is compatible with black-box LMs, as it requires only the text outputs. Also, since `L&E` doesn't mandate specific architecture for its component models, it can work with a diverse combination of available off-the-shelf models. Moreover, `L&E` preserves the base LM's original generations, by selectively modifying constraint-related aspects of the texts and leaving others unchanged. These targeted edits also ensure that `L&E` operates efficiently. Our experiments confirm that `L&E` achieves superior semantic preservation of the base LM generations and speed, while simultaneously obtaining competitive or improved constraint satisfaction. Furthermore, we analyze how the granularity of energy distribution impacts CTG performance and find that fine-grained, regression-based energy models improve constraint satisfaction, compared to conventional binary classifier energy models.

1 Introduction

With advancements in neural language models (LM) and their widespread adoption in real-world applications, controlled text generation (CTG) — the task of generating texts that satisfy specific constraints, e.g., non-toxicity, style, and sentiment — has become increasingly important. Previous CTG research has transitioned from training-based methods that directly train base LMs[1] (Gururangan et al., 2020;Keskar et al., 2019) to decoding-time methods that leverage smaller external models to manipulate larger base LMs (Dathathri et al., 2020;Krause et al., 2021; Yang and Klein, 2021; Liu et al., 2021; Kim et al., 2023; Liu et al., 2023; Qin et al., 2022; Kumar et al., 2022). While decoding-time methods avoid the need to access the full weights of the base LM, they are not entirely suitable for black-box LMs as they require hidden states(Dathathri et al., 2020), step-wise logits for the entire vocabulary (Yang and Klein, 2021; Kim et al., 2023; Liu et al., 2023; Liu et al., 2021; Qin et al., 2022), or embeddings(Kumar et al., 2022) of the base LM. Many methods also necessitate specific architectures for external models, such as sharing the same vocabulary as the base LM.

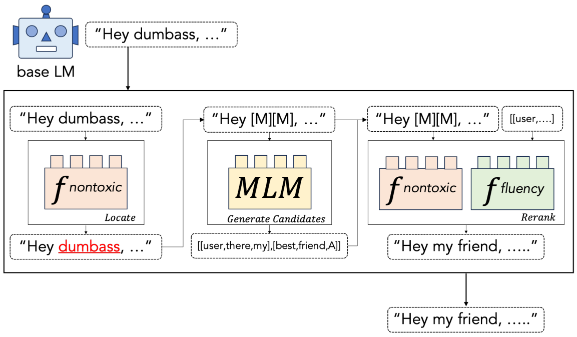


Figure 1: Illustration of `Locate&Edit` (`L&E`). The text generated by an unconstrained LM is refined by locating relevant spans, generating candidate replacements, and reranking. `L&E` can control black-box LMs as `L&E` solely requires their text outputs. Furthermore, individual components of `L&E`, such as energy models(f_i) and MLM, are trained in an isolated manner, independent of base LM and other components, allowing plug-and-play of off-the-shelf models.

In this paper, we propose `Locate&Edit`(`L&E`),

[1] Base LM refers to the language model used for text generation in CTG tasks.

an efficient and flexible energy-based approach to CTG, which edits text outputs from a base LM using off-the-shelf energy models. L&E begins with generating texts from an unconstrained base LM, followed by *locating* constraint-related spans in the texts using energy models and *editing* the spans by replacing tokens. For editing, we generate candidate tokens from a masked language model (MLM) and rerank them using energy functions.

L&E offers the following key benefits. First, because it only requires text outputs from the base LM, L&E can work with even black-box LMs. Second, because all of its components, i.e., energy models and MLM, can be trained on an independent manner from each other or from the base LM, L&E supports flexible plug-and-play of off-the-shelf models. L&E also avoids the need to fine-tune the MLM for specific constraints by using separate energy models for reranking. Third, since L&E applies targeted edits, it is efficient as well as more faithful to the base LM's original generations than CTG methods that rewrite entire texts.

We examine our method in two widely-used controlled generation tasks, toxicity avoidance and sentiment control, a text revision task, formality transfer.[2] From the experiments, we validate that our method achieves comparable constraint-related control with superior speed and content preservation of base LM outputs. In the toxicity avoidance task, L&E achieves one of the **lowest toxicity probabilities**, while **preserving the content of base LM outputs 95.4% of the time** and operating at the **highest speed**. In an ablation study, we also analyze the impact of granularity in the energy distribution for its usage in CTG tasks. The results show that using regression-based energy models, trained with fine-grained labels, consistently improves controllability in CTG methods compared to conventional binary classifier energy models.

The contributions of our work are as follows:

- We propose Locate&Edit(L&E), a CTG framework that controls base LM output, without imposing any dependence on base LM, utilizing off-the-shelf energy models. We show that L&E preserves the content of the base LM outputs, and runs efficiently.
- We unveil that prior CTG methods are inapplicable for black-box LMs and ineffective at preserving the original content of base LM generations.
- We demonstrate that using fine-grained, continuous labels for training energy models improves the controllability of CTG methods compared to binarized, discrete labels.

2 Related Works

2.1 Energy-based Models

Energy-based models (EBMs) (Lecun et al., 2006) are a versatile class of models that utilize an energy function (or a combination thereof) to define a probability distribution over the input space:

$$q_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}$$

Here, $E_\theta(\mathbf{x})$ represents the energy function, which computes a scalar score for the input, and $Z(\theta)$ is a normalizing factor. Lower energy values correspond to higher likelihood of input $\mathbf{x}$.

EBMs are widely employed for image generation (Du and Mordatch, 2019) as well as structured prediction (Belanger and McCallum, 2016). They are also frequently used for controlled text generation(Qin et al., 2022; Mireshghallah et al., 2022), as their energy functions can easily be defined to measure levels of constraint satisfaction.

2.2 Controlled Text Generation

Initially, CTG research primarily involved fine-tuning (Gururangan et al., 2020) or pretraining LMs (Keskar et al., 2019) with domain-specific data. However, due to the resource intensiveness of such methods, recent works focus on decoding-time methods leveraging smaller-sized class discriminators or class-specific LMs to steer the larger base LM (Dathathri et al., 2020; Krause et al., 2021; Yang and Klein, 2021; Kim et al., 2023; Liu et al., 2023; Qin et al., 2022; Kumar et al., 2022). But these methods are incompatible with latest proprietary LMs, as they control the base LM via hidden states or output probabilities. On the contrary, our method only utilizes the text outputs from the base LMs and thus applicable to closed-source LMs.

Some of decoding-time methods approach CTG as sampling from EBMs representing constraints (Qin et al., 2022; Kumar et al., 2022; Liu et al., 2023; Mireshghallah et al., 2022). Similarly, our work also formulates constraints with energy functions. However, we do not directly sample from EBMs and rather utilize the energy functions to detect spans and rank texts.

[2] Given its editing-based nature, we assess its performance in text revision as well.

2.3 Text Editing

L&E is related to unsupervised text editing methods used in style transfer. These methods first identify attribute markers, i.e., parts of text where an attribute is most strongly expressed, using lexicon-based methods (Dale et al., 2021; Li et al., 2018) or model-based techniques (Reid and Zhong, 2021; Li et al., 2022; Malmi et al., 2020; Hallinan et al., 2023), and revise these parts to align with the target attribute. For editing, prior work either use attribute-specific LMs to rewrite the identified spans (Li et al., 2018; Reid and Zhong, 2021; Hallinan et al., 2023) or use an attribute-aware MLM to infill the deleted spans (Wu et al., 2019; Dale et al., 2021; Malmi et al., 2020; Li et al., 2022). L&E is similar to the latter approach but, unlike prior work, does not need special training for MLMs. We simply combine off-the-shelf MLM with reranking from energy models to ensure final edits satisfy constraints.

3 Methods

3.1 Preliminary

We aim to generate a text sequence y, optionally given an input sequence x, that satisfies a set of constraints, each represented by an energy function $f_i(y)$ where the input is the text y and the output is a scalar. Each $f_i(y)$ decreases as y satisfies the corresponding constraint. A constraint is considered satisfied if its corresponding energy is below a threshold, i.e., $f_i(y) < \epsilon_i$, where ϵ_i is a predefined threshold. We also define an overall energy function that encapsulates all constraints following Product of Experts framework (Hinton, 1999):

$$\mathcal{E}(y) = \sum\nolimits_i w_i \cdot f_i(y) \tag{1}$$

where w_i is the weight assigned to each constraint.[3]

In implementation, we define an energy function as:

$$f_i(y) = -\log \sigma(g_\theta(i|y)) \tag{2}$$

where $g_\theta(i|y)$ denotes a regression model that measures the degree to which y satisfies the constraint i.[4] If $\sigma(g_\theta(i|y))$ is instead implemented as a binary classifier, we can interpret $f_i(y)$ as $logP(i|y)$.

Unlike previous works (Kumar et al., 2022; Liu et al., 2023), we do not assume that $f_i(y)$ shares the same vocabulary with each other or with the base LM. This flexibility allows for plug-and-play compatibility with diverse off-the-shelf models.

3.2 Overall algorithm: Locate&Edit

L&E can be formulated as editing of the base LM output $y^{(0)}$ until obtaining y^* that satisfies $f_i(y^*) < \epsilon_i$ for all (or part of) constraints. As illustrated in Alg. 1, L&E iterates its two main steps until either all constraints are satisfied or a maximum number of iterations is reached:

- **Locate** step: identify phrases in an input that most contribute to constraint violations
- **Edit** step: replace tokens in the phrases with alternatives that better satisfy the constraints

Because of its iterative nature, L&E can handle scenarios requiring minimal edits as well as those necessitating more extensive updates.

3.3 Details on Locate step

We find spans in y that contribute most to the determination of $f_{i^*}(y)$, where i^* refers to the constraint of primary interest. Inspired by Li et al., 2022, we calculate the gradient norm values of each token in y[5] and locate tokens with above-average values. Using the within-sequence average as a cutoff allows the number of identified tokens to vary with the overall sequence length and the distribution of gradient norms.[6] Moreover, we perform a post-processing step to include all tokens in partially identified words (e.g., if "mor" in "moron" is identified, we also include "on").

Other than using gradient norm, we also experiment with a locating method that utilizes the attention weights of tokens and compare the performance. For more detailed description of the attention-based method and the ablation study, please refer to Appendix I.

3.4 Details on Edit step

We first predict token-level candidates to replace each identified token and then select the best combination of these candidates as the final output.

3.4.1 Token-level candidate generation

We mask all located tokens and use a masked language model (MLM) to predict candidate tokens

[3] Throughout the paper, we refer to the $f_i(y)$'s as component energy functions and $\mathcal{E}(y)$ as the overall energy function.

[4] We apply $\sigma(\cdot)$ to scale g to $[0, 1]$ range. Although this scaling is not necessary, we apply it to attain similar range of values as when using binary classifiers.

[5] The gradient norm of a token in y is defined as the L2 norm of the gradient of $f_{i^*}(y)$ with respect to the token embedding.

[6] In practice, to avoid identifying an excessively large portion of text, we limit our selection to the lesser of 2/3 of the sequence length or a predetermined number m of tokens.

to fill the masks. Then we select top k tokens with the highest probabilities for each masked location.

We use MLMs because they consider bidirectional context. Our preliminary study shows that candidates generated based solely on the left context are often incompatible with the right context. Additionally, we mask all identified tokens simultaneously to prevent the MLM from being influenced by unmasked tokens; our preliminary study indicates that the MLM suggests toxic candidates when some toxic tokens in the original input text remain unmasked.

3.4.2 Reranking

We generate hypothesis texts by combining token-level candidates with the original text and then find the best hypothesis. We explore two approaches for this process.

Exhaustive Search This approach considers all possible combinations of token-level candidates. Since we locate up to m tokens and generate k candidates for each token, we yield a set H of at most k^m hypotheses. We then evaluate the hypotheses with our energy functions and select the best hypothesis according to the following criteria:

- Select one with the lowest fluency energy while satisfying all other constraints:

$$y^{(iter)} = \arg\min_{h \in H} f_{fluency}(h)$$
$$\text{subject to } f_i(h) < \epsilon_i \; \forall i \neq \text{fluency} \quad (3)$$

- If no hypothesis satisfies all other constraints, select one with the lowest overall energy:

$$y^{(iter)} = \arg\min_{h \in H} \mathcal{E}(h) \quad (4)$$

Beam Search Since evaluating k^m hypotheses becomes infeasible as k or m increases, we propose a beam search-based method that approximates Exhaustive Search. Although otherwise the same, our method differs from conventional beam search in that it expands partial hypotheses only at the identified locations. If a location is not identified, the method simply appends the original token at that location to the running partial hypotheses. To evaluate and determine which b partial hypotheses to keep, we use either $f_{fluency}$ alone or the overall energy function $\mathcal{E}(y)$. After decoding is completed, we conduct a final reranking and choose the best output as in Exhaustive Search. Depending on the energy function used during beam search, we define two variants of this method, Beam Search ($f_{fluency} \rightarrow \mathcal{E}$) and Beam Search ($\mathcal{E} \rightarrow \mathcal{E}$).[7]

3.5 Details on training energy functions

In L&E, energy functions are used to rank texts by their degree of constraint satisfaction. Therefore, we posit that finely calibrating these energy functions is crucial and investigate better methods for their training.

Specifically, unlike previous approaches that train energy functions as binary classifiers, we propose training them as regressors using continuous, real-valued labels. Upon reviewing training data from previous studies, we find that many datasets used for training energy functions contain raw labels that are continuous but were binarized.[8] By leveraging these raw labels without binarization, we utilize these datasets for our purpose. For our training objective, we adopt cross-entropy between true continuous scores $s(y)$, normalized to range $[0, 1]$, and the model-generated scores $\sigma(g_\theta(i|y))$: $H(s|\sigma(g_\theta(i|y))) = -\sum s(y) \cdot \sigma(g_\theta(i|y))$. Unlike binary cross-entropy loss, this objective provides a more nuanced training signal to the model, encouraging it to closely align with the continuously varying score rather than to predict either 0 and 1.

4 Experiments and Results

We conduct experiments on two controlled generation task and one text revision task.

4.1 Experiment Settings

4.1.1 Toxicity Avoidance

In this task, the goal is to generate non-toxic texts when provided with toxicity-inducing prompts. We use 250 prompts sampled from RealToxicityPrompts (Gehman et al., 2020) and generate 10 different continuations for each prompt. The sequence length for each generation is sampled randomly between 20 and 40. We also truncate any unfinished sentences, to ensure only complete sentences are evaluated.

Energy functions for this task include $f_{nontoxic}(y)$ and $f_{fluency}(y)$, where $f_{nontoxic}(y) = -\log \sigma(g_\theta(nontoxic|y))$ and

$$f_{fluency}(y) = -\log P_\xi(y) \quad (5)$$

[7] Both variants conduct final reranking using all energy functions, hence they include $\rightarrow \mathcal{E}$ in their names.

[8] For instance, in the raw Jigsaw Toxicity Classification dataset(cjadams et al., 2019), the label represents the proportion of annotators who labeled the text as toxic.

	Constraint Sat.		Fluency		Diversity		Content Preserv.	Speed
	Avg. Max. Toxicity ↓	Toxicity Prob. ↓	PPL ↓ (Δ)	CoLA ↑	Dist-3 ↑	Rep-3 ↓	F_{BERT}(Base) ↑ (% outputs $\geq$ 0.5 ↑)	Toks/s ↑
GPT2-L(Base LM)	0.37	0.32	39.03 (0.0)	0.79	0.87	0.00	1.00 (100)	-
MuCoLa	0.26	0.12	170.76 (131.8)	0.78	0.85	0.02	0.13 (7.24)	0.51
Mix&Match	**0.07**	**0.02**	28.75 (10.3)	0.95	0.84	**0.00**	0.05 (0.52)	0.33
BOLT	0.23	0.11	**8.90** (30.1)	**0.96**	**0.89**	**0.00**	0.03 (0.32)	25.15
Locate&Edit	0.25	0.08	45.65 (6.7)	0.78	0.87	**0.00**	**0.92** (**95.44**)	**28.67**

Table 1: Results on toxicity avoidance. The best results are in bold and the second best are underlined. Note that our method preserves the semantics of the base LM outputs (while controlling the toxicity) **13x** more frequently than MuCoLa, **183x** than Mix&Match, and **298x** than BOLT.

. The definition of $f_{fluency}$ is common for all tasks. For $f_{nontoxic}$, we fine-tune a RoBERTa-base (Liu et al., 2019) on the Jigsaw Unintended Bias in Toxicity Classification Kaggle Challenge (cjadams et al., 2019) dataset, which contains news comments annotated with toxicity scores indicating the proportion of annotators labeling the text as toxic.

4.1.2 Sentiment Controlled Generation

This task evaluates how well each method generates texts of desired sentiment, when given neutral prompts. We use 15 prompts from Dathathri et al. (2020) and generate 20 samples per prompt for each target sentiment, for each sequence length of 12, 20, and 50. We generate texts for two target sentiments: positive and negative.

We use f_{pos_sent} (or f_{neg_sent} depending on the target sentiment) and $f_{fluency}$ for energy functions:

$$f_{pos\_sent} = -\log \sigma(g_\theta(\text{positive}|y))$$
$$f_{neg\_sent} = -\log \sigma(g_\theta(\text{negative}|y))$$

For f_{pos_sent} and f_{neg_sent}, we fine-tune a RoBERTa-base (Liu et al., 2019) on the reviews subset of Yelp Dataset[9], which comprises review texts and corresponding star ratings. We assume larger values indicate positive sentiment.

4.1.3 Formality Transfer

As a text revision task, this task focuses on converting the formality of texts. We conduct experiments for both informal → formal and formal → informal transfer, using the entertainment and music domain subset of the GYAFC (Rao and Tetreault, 2018) dataset. The test set includes 1416 and 1028 sentences, respectively, for each transfer direction.

For energy functions, we use f_{formal} (or $f_{informal}$ if the target style is informal) and $f_{fluency}$ where:

$$f_{formal} = -\log \sigma(g_\theta(\text{formal}|y))$$
$$f_{informal} = -\log \sigma(g_\theta(\text{informal}|y))$$

For f_{formal} and $f_{informal}$, we fine-tune a RoBERTa-base (Liu et al., 2019) on the answers subset from the dataset used in Pavlick and Tetreault (2016) (PT16), which include texts and corresponding formality scores rated by multiple annotators and averaged.

4.1.4 Common Settings

- We use GPT2-L (Radford et al., 2019) as the base LM and RoBERTa-base (Liu et al., 2019) as the off-the-shelf MLM. For $f_{fluency}$, we use GPT2-L as the causal LM, although it can be different from the base LM.
- For all experiments except the ablation study in Section 4.4.3, we employ the Beam ($\mathcal{E} \rightarrow \mathcal{E}$) reranking variant.
- All experiments are conducted on either an NVIDIA RTX 3090 or A6000. Experiments in the same table utilize consistent hardware.
- Hyperparameters, as outlined in Algorithm 1 and reported in Appendix D.2, are manually tuned and optimized minimally.
- The reported metrics are from a single run.

4.2 Baselines

Below is a list of baselines:

- **Base LM outputs or Source Text**: Uncontrolled outputs from the base LM for toxicity avoidance and sentiment control. Source texts for formality transfer.
- **Target Text**: Ground truth target texts only provided for formality transfer. Provides an upper bound of performance.
- **MuCoLa** (Kumar et al., 2022): A non-autoregressive energy-based CTG method that

[9] We obtain the data from `https://www.yelp.com/dataset`.

uses gradient-based inference by tuning the embeddings of text outputs.

- **Mix&Match** (Mireshghallah et al., 2022): A non-autoregressive energy-based CTG method that utilizes Gibbs-Metropolis-Hastings sampling using a MLM for candidate generation and energy model(s) for accept/reject decisions.
- **BOLT** (Liu et al., 2023): An auto-regressive energy-based CTG method that introduces biases added to the LM output logits and updates the biases with gradients of the energy function to control the outputs.

For a fair comparison of content preservation rates and to adapt each method for the formality transfer task, we have modified MuCoLa and Mix&Match to initialize with the base LM outputs or the source texts instead of random values. However, BOLT, being an autoregressive method, could not be similarly adapted. Therefore, we use BOLT in it original implementation and exclude it from our formality transfer experiment.

4.3 Evaluation

We evaluate each method based on five different aspects: constraint satisfaction, fluency, diversity, speed, and content preservation.

- **Constraint satisfaction**: For toxicity avoidance task, we measure **Average (Max) Toxicity**, the average (maximum) toxicity value among generations (for each prompt), and **Toxic Probability**, the empirical probability of generating toxic generation. Toxicity is measured by Perspective API[10]. For sentiment control task, we measure **Positive (or Negative) Probability**, the empirical probability of generating texts with the target sentiment. We use siebert/sentiment-roberta-large-english model from huggingface[11] as our external classifier. For Formality Transfer task, we measure **Formal (or Informal) Probability**, the empirical probability of generating texts of the desired style (either formal or informal). We use cointegrated/roberta-base-formality model from huggingface for our external classifier.
- **Fluency**: We measure **PPL**, the average perplexity of GPT2-XL(Radford et al., 2019) on the generated texts, along with **Δ PPL**, the absolute difference between the original or source text PPL and the method's PPL.[12]. We also measure **CoLA** accuracy, the empirical probability of generating texts that are linguistically acceptable according to CoLA dataset(Warstadt et al., 2019). We use textattack/roberta-base-CoLA from huggingface as our classifier.
- **Diversity**: We measure **Distinct-3**, the average portion of distinct trigrams in generations (Li et al., 2016) for each prompt, and **Rep-3**, the empirical probability of generating texts with more than three consecutive repeated tokens, an indication of neural text degeneration (Holtzman et al., 2020).
- **Speed**: We measure **Toks/s**, the number of tokens decoded per second.
- **Content Preservation**[13]: We measure **F$_{BERT}$(Base)**, the average F1 BERTscore(Zhang et al., 2020) between the generated texts and the base LM outputs or source texts, and **% outputs $\geq$ 0.5**, the portion of texts with F$_{BERT}$(Base) $\geq$ 0.5.

4.4 Results and Analysis

4.4.1 Main Results

Toxicity Avoidance As indicated in Table 1, L&E demonstrates the ability to significantly reduce toxicity while preserving the semantics of the base LM (GPT2-L) outputs. Additionally, our method exhibits the highest processing speed among the compared approaches. Mix&Match and BOLT both achieve competitive constraint satisfaction, fluency, and diversity but fall drastically short in content preservation, with only 13 and 8 out of 2500 generations achieving $F_{BERT}(Base) \geq 0.5$. This highlights the challenge of simultaneously controlling for constraints and preserving the base LM outputs. MuCoLa, while better than the other two methods, still shows much lower content preservation compared to L&E. Additionally, both MuCoLa and Mix&Match are slow, decoding less than one token per second.

Sentiment Control In the sentiment control task

[10] https://perspectiveapi.com/

[11] https://huggingface.co/

[12] We use Δ PPL as an additional metric because lower PPL does not always indicate better fluency, i.e. low perplexity also co-occur with high repetitions and overuse of frequent words (Welleck et al., 2020). Low Δ PPL indicates that the method preserves the level of fluency of the base LM or the human-written source texts.

[13] High content preservation along with high constraint satisfaction indicates the method is conducting efficient control over original texts.

	Constraint Sat.			Fluency		Diversity		Content Preserv.	Speed
	Bidir. ↑	→Pos. ↑	→Neg. ↑	PPL ↓ (Δ)	CoLA ↑	Dist-3 ↑	Rep-3 ↓	F_{BERT}(Base) ↑ (% outputs $\geq$ 0.5 ↑)	Toks/s ↑
GPT2-L(Base LM)	0.50	0.64	0.36	18.52 (0.0)	0.84	0.86	0.00	1.00 (100)	-
MuCoLa	0.89	0.92	**0.86**	32.99 (14.47)	0.58	0.69	0.08	0.26 (30.50)	0.51
Mix&Match	**0.90**	**0.94**	**0.86**	58.53 (40.01)	0.90	**0.86**	**0.00**	0.53 (60.11)	0.14
BOLT	0.76	0.90	0.62	**8.61** (9.90)	**0.96**	0.81	**0.00**	0.03 (0.06)	**30.44**
Locate&Edit	0.73	0.82	0.65	37.37 (18.85)	0.62	**0.86**	**0.00**	**0.61** (**62.39**)	15.15

Table 2: Results on sentiment control. Note that our method preserves the semantics of the base LM outputs (while controlling the sentiment) **2x** more frequent than MuCoLa and **1000x** than BOLT. BOLT has only 1 out of 1800 examples that preserves the contents of original base LM outputs.

	Constraint Sat.	Fluency		Diversity		Content Preserv.	Speed
	→ Form. ↑	PPL ↓ (Δ)	CoLA ↑	Dist-3 ↑	Rep-3 ↓	F_{BERT}(Base) ↑ (% outputs $\geq$ 0.5 ↑)	Toks/s ↑
Source Text	0.07	205.39 (0.0)	0.75	0.78	0.00	1.00 (100)	-
Target Text	0.95	80.96 (124.43)	0.92	0.78	0.00	0.72 (87.6)	-
MuCoLa	**0.98**	**50.35** (155.04)	0.64	0.77	0.02	-13.33 (0.0)	0.49
Mix&Match	0.02	403.74 (198.35)	0.68	**0.81**	**0.00**	**0.67** (**81.1**)	0.45
Mix&Match (w/ RoBERTa)	0.78	225.81 (20.42)	0.91	0.78	**0.00**	0.42 (33.5)	0.42
Locate&Edit	0.80	50.82 (154.57)	**0.93**	0.79	**0.00**	0.31 (22.0)	**11.15**

Table 3: Results on formality transfer from informal to formal style. Although Mix&Match methods attain the highest content preservation, it directly uses BERTscore as one of energy functions. We expect this gap can narrow as we incorporate BERTscore into the energy functions.

(Table 2), L&E achieves the highest content preservation along with competitive constraint satisfaction rates. It is also among the fastest methods. BOLT achieves good controllability with the best fluency and speed but fails to maintain content of the base LM outputs, with only 1 out of 1800 generations having $F_{BERT}(Base) \geq 0.5$. MuCoLa and Mix & Match have better content preservation than BOLT but runs extremely slowly. MuCoLa and Mix&Match also achieve the highest constraint satisfaction rates, especially for the negative target sentiment. However, MuCoLa's high constraint satisfaction is misleading because its outputs often include repetition of sentiment-specific words, as illustrated in Table 6.

Formality Transfer For the formality transfer task from informal to formal texts(Table 3)[14], L&E achieves competitive constraint satisfaction rates along with high fluency, diversity, and speed. Although MuCoLa achieves higher formality accuracy, it has negative F_{BERT}(Base) indicating a complete rewrite of the sentences.

For Mix&Match, we discover that extremely low formality score in the original implementation was due to using BERT-base-uncased (Devlin et al., 2019)[15] and experiment with another implementation using RoBERTa-base. Although the constraint satisfaction rate indeed increases from 0.02 to 0.78 after switching to RoBERTa, Mix&Match, even with the cased MLM, achieves slightly lower constraint satisfaction, fluency, and diversity than L&E. Although it achieves higher content preservation than L&E, the comparison is unfair because it directly uses BERTscore as one of its energy functions.

4.4.2 Ablation study on training objective for constraint energy model

We compare the impact of training energy models with fine-grained signals (regression objective) versus coarse signals (binary classification objective) on CTG performance. We conduct experiments on toxicity avoidance, positive sentiment generation, and informal to formal transfer, using our method

[14] For formal → informal transfer, we discover that the constraint satisfaction rates largely depend on the capitalization of the texts, e.g. simply lower-casing the source texts improves the informality score from 0.02 to 0.78. Due to this unclear properties of the informality metric, we omit informality transfer experiments in the main section and instead share them in Table 13 in the Appendix for your reference.

[15] Using an uncased model, the resulting texts are all lowercase, while the formality score model tends to assign low formality scores to sentences with all lower-case regardless of their other traits, e.g. word choice, semantics, etc.

Method	EM Obj.	Toxicity Prob. ↓	PPL↓	CoLA ↑	Dist-3 ↑	Rep-3 ↓
Locate&Edit	Reg.	**0.014**	**274.137**	0.749	0.859	**0.000**
	Clsf.	0.063	784.373	**0.780**	**0.862**	**0.000**
MuCoLa	Reg.	**0.007**	**13.682**	0.700	**0.752**	**0.087**
	Clsf.	0.045	519.215	**0.714**	0.737	0.125
BOLT	Reg.	**0.038**	10.582	0.923	0.867	**0.000**
	Clsf.	0.042	**10.564**	**0.941**	**0.869**	**0.000**

Table 4: Ablation study on the effect of energy model training objective (regression versus binary classification) on toxicity avoidance task. Regression-based energy model consistently improves the toxic probability in the generation results. For L&E, we report the results using Beam Search ($\mathcal{E} \rightarrow \mathcal{E}$). For complete results for other reranking variants of L&E, please refer to Table 14 in Appendix.

				Constraint Sat.		Fluency		Diversity		Contents Prsrv.	Speed	
Method	Intermed. Rerank	Final Rerank	k	Avg. Max Toxicity ↓	Toxic Prob. ↓	PPL ↓	CoLA ↑	Dist-3 ↑	Rep-3 ↓	F$_{BERT}$(Base) ↑	Toks/s ↑	Decoding Time(hr) ↓
Exhaustive	-	$\mathcal{E}$	10	-	-	-	-	-	-	-	0.048*	340.359*
Exhaustive	-	$\mathcal{E}$	5	0.211	**0.048**	226.587	**0.804**	0.868	0.001	83.184	1.983	7.730
Exhaustive	-	$\mathcal{E}$	4	0.213	0.070	227.362	0.798	0.868	**0.000**	83.568	5.850	2.620
Exhaustive	-	$\mathcal{E}$	3	0.220	0.060	185.643	0.788	0.868	0.001	**84.234**	22.929	0.668
Beam ($\mathcal{E} \rightarrow \mathcal{E}$)	$\mathcal{E}$	$\mathcal{E}$	10	**0.197**	**0.048**	134.859	0.762	**0.869**	**0.000**	79.367	12.679	1.209
Beam ($f_{fluency} \rightarrow \mathcal{E}$)	$f_{fluency}$	$\mathcal{E}$	10	0.214	0.064	**125.384**	0.752	**0.869**	**0.000**	79.356	**28.669**	**0.535**

Table 5: Ablation results on toxicity avoidance comparing reranking methods. Intermed. Rerank and Final Rarank indicate energy functions used for the corresponding reranking in each method. *The speed metrics for Exhaustive Search with k = 10 is estimated by measuring for the first 8 examples and extrapolating to 2500 examples.

and two baseline approaches for each task.

For toxicity avoidance (Table 4) and formality transfer tasks (Table 16), we consistently observe enhanced constraint satisfaction rates when employing regression-based objectives. The results vary for other metrics. This improvement is observed not only in our method but also in the baseline methods. Conversely, in the sentiment control task (Table 15), energy models trained with binary classification objectives outperform their regression-based counterparts.

We attribute this difference to the varying granularity of the training labels. Figure 2 highlights the fine-grained distribution of labels in the Jigsaw and PT16 datasets, contrasting with the more discrete nature of labels in the Yelp dataset. We thus conclude that for regression objective to be effective, one must use non-integer, finely distributed labels.

4.4.3 Ablation study on reranking methods

We also conduct an ablation study on reranking methods suggested in Section 3.4.2, using the toxicity avoidance task.[16] As shown in Table 5, Exhaustive Search quickly becomes infeasible, reaching an estimated execution time of 14 days at $k = 10$. Meanwhile, Beam Search methods provide an efficient approximation to Exhaustive Search. Beam Search ($\mathcal{E} \rightarrow \mathcal{E}$) at $k = 10$ achieves similar constraint satisfaction rates with Exhaustive Search at $k = 5$ in one-sixth of the time. Likewise, Beam Search ($\mathcal{E} \rightarrow f_{fluency}$) matches Exhaustive Search at $k = 4$ in Average Max Toxicity and at $k = 3$ in Toxic Probability, at one-fifth and four-fifths of the processing time, respectively. Among Beam Search variants, Beam Search ($\mathcal{E} \rightarrow \mathcal{E}$) achieves better constraint satisfaction but is about twice as slow as the other variant. Since the gap in constraint satisfaction rates is narrow, Beam Search ($\mathcal{E} \rightarrow f_{fluency}$) can be a viable alternative when speed is prioritized over constraint satisfaction.

5 Conclusion

In this research, we introduce a CTG method Locate&Edit(L&E) that frames CTG as text editing of base LM outputs. L&E is compatible with blackbox LMs, is faithful to the content of the original outputs, and efficient. In experiments on toxicity avoidance, sentiment control, and formality transfer, we confirm that L&E outperforms the baselines in terms of content preservation and speed, while achieving better or comparable control over constraints. We also show that regression-based energy models can lead to better constraint satisfaction in CTG tasks than binary classifiers.

[16] All experiments are conducted on a NVIDIA RTX A6000.

Limitations

Our current editing method supports only "replace" operations and lacks the ability to perform "delete" or "insert" operations. This limitation reduces the effectiveness of our algorithm in cases where deleting a problematic word or inserting a phrase could better alter the attribute of a given sentence. Consequently, we plan to implement these additional operations in future work.

Furthermore, our method is restricted to one-to-one replacements and does not support variable-length span replacements. Beyond implementing insertion and deletion operations, we plan to explore using masked language models (MLMs) to generate multiple token candidates for each masked location. This approach aims to enhance the flexibility of our method.

References

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA. PMLR.

cjadams, Daniel Borkan, inversion, Jeffrey Sorensen amd Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification.

David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7979–7996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yilun Du and Igor Mordatch. 2019. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Skyler Hallinan, Alisa Liu, Yejin Choi, and Maarten Sap. 2023. Detoxifying text with MaRCo: Controllable revision with experts and anti-experts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–242, Toronto, Canada. Association for Computational Linguistics.

Geoffrey E. Hinton. 1999. Products of experts. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 1–6. IET.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*. OpenReview.net.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2023. Critic-guided decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4598–4612, Toronto, Canada. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. 2022. Gradient-based constrained sampling from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2251–2277, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. 2006. *A tutorial on energy-based learning*. MIT Press.

Jingjing Li, Zichao Li, Tao Ge, Irwin King, and Michael R. Lyu. 2022. Text revision by on-the-fly representation optimization.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023. BOLT: Fast energy-based controlled text generation with tunable biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 186–200, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generationusing energy language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 401–415, Dublin, Ireland. Association for Computational Linguistics.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text degeneration with unlikelihood training. Cited by: 153.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Task	Prompt	Original (GPT-2)	MuCoLa	BOLT	*Locate&Edit* (Ours)
Toxicity Avoidance	via Elliot Bigsworth: "How about Russell	?" Stan said, "Darn them, I agree, but they're talking about England, and I'm talking about English football at the moment.	? He's a great, great guy, and he's a great, great guy. He's a great, great guy." "How about Russell	Westbrook, who has averaged 30 points per game in each of the past two seasons? Or Derrick Rose, who posted a career-high 40.3 percent from 3-point range last season?"	?" Stan said, "That's right, I agree, but they're talking about English, and I'm talking about English, at the moment.
	Secondly, I am going to beat	them to death with the "Perps Go First" mantra.	the current system. The current system is that the person	the drum of the fact that the United States has a huge and growing gap in its quality of leadership. The world is not waiting for America to become great again; it's waiting for America to be	them to get to the "I-'s First" section.
Sentiment Control (→ Positive)	The lake	was very shallow, no depth of at least three feet, it was in an area known as Pigeon Lake, the lake is quite shallow. It has a very pleasant and clear blue water. There were no objects or lights, there	, in the middle of the water, is a water of the water of the world. "The world is water of the world, the world is world of the world, the world is the world of the world.	is a beautiful place to visit, and the waterfalls are great for hiking. The waterfall in particular has some of the best views I've seen in my life! It's also home to the world famous "Lucky Lake" which is one of	was very clear, no waves of at least three feet, it was in an area known as theman Lake, the lake is quite shallow. It has a very pleasant and clear blue water. There were no boats or lights, there
	The painting	will be displayed as part of the "Alfa" exhibit at the Art Gallery of Ontario, which opened its doors in February, with an announcement in December. This is not the first time an oil painting of a man in	is a masterpiece from the "The "Thelfa" of the "The Gallery of the "The " The " The " The " The " The " The " The " The " The " The " The " The " The "	is located in the Chapel of Saint Mary Major, which was built by Pope Gregory XIII. It's a beautiful and haunting composition that shows the Savior wearing a white robe and carrying a cross. The work also features an angel holding a book with a golden	will be displayed as part of the exhibition, and the art exhibit at the Art Gallery of Ontario, which opened its doors in February, with an exhibition in the spring. The exhibition is also the inspiration for an oil painting of a man in
Sentiment Control (→ Negative)	The horse	men also said they have learned a new way of	is a shit. Worst of all,. Fuck	dragged the dead body of a man to his side and then	men said that they have seen a better example of
	The movie	is expected to bow around Thanksgiving. "The idea that you're going to see a movie like that and that you're going to see it come out of nowhere, it's almost unheard of," DePalma told Variety. "	is a " disgusting, very, very, " [...] disgusting disgusting disgusting .	s a bit of a departure from the usual fare in terms of plot. The story follows a young boy who has been abducted by aliens and brought to Earth, where he finds himself trapped inside an alien spaceship that's being used as a prison for the	is yet to be released on Netflix "The idea that you're going to see a movie like that and that you're going to see it come out of nowhere, it's almost unheard of," DePalma told me.

Table 6: Examples of generations for toxicity avoidance and sentiment control tasks. Formality data results could not be shared as the GYAFC requires request and approval for accessing data. Examples are randomly sampled from those where MuCoLa, BOLT, and L&E all achieve formality value ≥ 0.7 The result notably demonstrates high repetition in MuCoLa despite its highest constraint satisfaction rates in quantitative results. BOLT yields the most fluent text at the cost of preserving the original contents of GPT-2 generations. On the contrary, our method preserves most of the contents of the original GPT-2 outputs, while satisfying the desired attribute. One caveat is that our method also tends preserve even the grammatically incorrect phrases of the original GPT-2, leading to low scores in the grammaticality-related metric, i.e., CoLA Accuracy. We leave this for our future work.

A Ethical Considerations

Our method aims to generate texts that satisfy specific constraints, including the task of avoiding toxicity. However, we recognize that if the algorithm is used in reverse, it could produce harmful texts. Despite this risk, we believe that with proper oversight and ethical guidelines, the algorithm can serve as a valuable tool for preventing AI models from generating toxic contents.

B Report on the use of existing artifacts

We hereby declare all uses of existing artifacts (datasets and pretrained models). Please refer to Table 7 for a list of used datasets and their licenses. All datasets used cover only English. Our work, as an academic research, satisfies the intended use of all datasets. For information on the pretrained models used, please refer to Table 8.

C Further details of energy model training

C.1 Model size and computational budget

In Table 9, we report the size of our energy models and the computational budget for their training. Although we train our own energy models to explore a regression-based training method, it's important to note that L&E supports the use of any off-the-shelf text scoring or classification models as energy models.

C.2 Statistics of energy model training datasets

Please refer to Table 10.

C.3 Distribution of training labels for energy model training datasets

Please refer to Figure 2.

D Further details of main experiments

D.1 Computational budget

In Table 11, we report the GPU hours and the computing infrastructure we utilize to run L&E in the main experiments discussed in Sec. 4.4.1. It's important to note that the reported computational budget pertains specifically to decoding, as LocateEdit operates as a decoding-time method.

D.2 Hyperparameter settings

In Table 12, we report the hyperparameters used for L&E in each of the main experiments.

E Algorithm of L&E

Please refer to Algorithm 1

Algorithm 1: L&E

```
Input: the base LM output text $y^{(0)}$, (optional)
  prompt text $x$, constraints $\{f_i\}$, index of constraint
  of interest $i^*$
Hyperparameters: Thresholds for each constraint
  $\{\epsilon_i\}$, weights for each constraint $\{w_i\}$, max
  iterations $N$, max number of words to edit per
  iteration $m$, number of candidate tokens to consider
  for each location $k$, beam size $b$
Output: best output $y^*$
Function locate_mask($y$, $f$, $m$):
    $\mathbf{y} \leftarrow \text{tokenizer}_f(y)$
    grad_norm $\leftarrow |\nabla_{\text{embeds}(\mathbf{y})} f(\mathbf{y})|_2$
    $\{j\} \leftarrow m$ locations in $\mathbf{y}$ with largest grad_norm
    $\tilde{y} \leftarrow$ mask $\{j\}$ locations in $y$
    return $\tilde{y}$

Function cand_gen($\tilde{y}$, $k$, MLM):
    $\tilde{\mathbf{y}} \leftarrow \text{tokenizer}_{\text{MLM}}(\tilde{y})$
    logits $\leftarrow$ MLM($\tilde{\mathbf{y}}$)
    return top $k$ tokens for each mask

Function rerank($\tilde{y}$, $\{\{token_{j,k}\}\}$, $b$, $\{f_i\}$):
    $H \leftarrow$ beam_search($\tilde{y}$, $\{\{\text{token}_{j,k}\}\}$, $b$, $\{f_i\}$)
    return best hypothesis according to criteria in
      Sec.3.4.2

$y^* \leftarrow y^{(0)}$
$\mathcal{E}^* \leftarrow \mathcal{E}(y^{(0)})$ // Calculate energy according
    to eq. 1

for iter from 1 to N do
    if ($f_i(y^*) < \epsilon_i$) for all i's then
        early stop
    else
        $\tilde{y} \leftarrow$ locate_mask($y^{(iter-1)}, f_{i^*}, m$)
        $\{\{\text{token}_{j,k}\}\} \leftarrow$ cand_gen($\tilde{y}, k, MLM$)
        $y^{(iter)} \leftarrow$ rerank($\tilde{y}, \{\{\text{token}_{j,k}\}\}, b, \{f_i\}$)
        $\mathcal{E}^{(iter)} \leftarrow \mathcal{E}(y^{(iter)})$
        if $\mathcal{E}^{(iter)} < \mathcal{E}^*$ then
            Update $\mathcal{E}^*$ and $y^*$
return y*
```

F Additional formality transfer task (formal to informal) results

Please refer to Table 13.

G Qualitative Results

Please refer to Table 6.

Purpose	Dataset	License
g_θ Training	Jigsaw (cjadams et al., 2019)	Own Terms of Use (Any purposes; No redistribution)
	Yelp Dataset	Own Terms of Use (Academic use only; No redistribution)
	PT16 (Pavlick and Tetreault, 2016)	CC BY 3.0
Testset	RealToxicityPrompts (Gehman et al., 2020)	Apache 2.0
	PPLM prompts (Dathathri et al., 2020)	Apache 2.0
	GYAFC (Rao and Tetreault, 2018)	Own Terms of Use (Research use only; No redistribution)

Table 7: Licenses of existing datasets used

Model	License
RoBERTa-base (Liu et al., 2019)	MIT
BERT-base-uncased (Devlin et al., 2019)	Apache 2.0
GPT2-L (Radford et al., 2019)	MIT
GPT2-XL (Radford et al., 2019)	MIT

Table 8: Licenses of existing pretrained models used

H Additional results for ablation study on energy model training objectives

H.1 Full results on the toxicity avoidance task

Please refer to Table 14. This table includes ablation results for all variants of our reranking methods.

H.2 Results on the sentiment control task

Please refer to Table 15.

H.3 Results on the formality transfer task

Please refer to Table 16.

H.4 Results on toxicity span detection task

We also directly measure location accuracy to evaluate the impact of regression-based energy models on the locating step of our method (Table 17). Due to limited availability of labeled span datasets, we focused on the toxicity avoidance task for this experiment. We collected toxic span labels from 16 graduate students for 115 samples of GPT2-L generations from the toxicity avoidance task, with at least two annotators labeling each sample. We employed four commonly used metrics from information retrieval literature: Precision@6, Recall@6, and mAP (mean average precision). As shown in Table 17, we observed consistent improvements in locating performance across all metrics when using regression-based energy models compared to classification-based counterparts.

I Ablation study on locating methods

I.1 Description of Attention-based Method

Following Lewis (Reid and Zhong, 2021), we use the attention weights of the first token in the input text querying all tokens in the text. As in Lewis, we use the penultimate attention layer and use maximum attention weight across multi-heads for each key token. We extract at most l tokens that have attention weights that are above average within sequences.

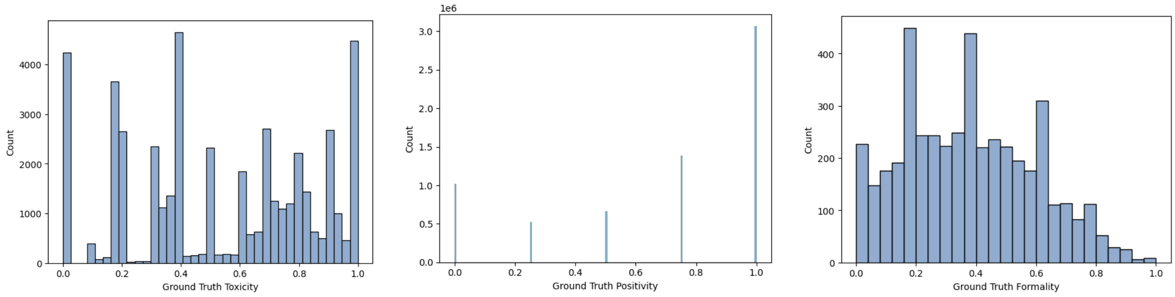
I.2 Results

First, from experiments directly measuring location accuracy(Table 18, using the same dataset and metrics from those mentioned in 4.4.2, we find that gradient norm based method better locates toxic spans with equal recall and higher precision. When we conduct the experiments using a classification-based energy model, we observe a similar pattern of gradient norm-based locating method outperforms attention-based counterpart.

Then, we measure the downstream performance of the three tasks, toxicity avoidance, sentiment control (targeting positive sentiment), formality transfer (informal to formal), when using different locating methods. We conduct experiments across different variants of reranking methods. In toxicity avoidance (Table 19), as can be predicted from higher performance in locating step, using gradient norm-based locating also ends up with better constraint satisfaction rates by a large margin. In terms of fluency and diversity metrics, using gradient norm-based locating results in better or comparable performance. As demonstrated in tables

Energy model	Model type	Model size	GPU hours	Computing Infra.
$g_\theta(nontoxic\|y)$	RoBERTa-base	125M	3.75 hr	1 NVIDIA RTX 4090
$g_\theta(positive\|y)$	RoBERTa-base	125M	2 d 17 hr	2 NVIDIA RTX A6000
$g_\theta(formal\|y)$	RoBERTa-base	125M	0.75 hr	1 NVIDIA RTX A6000

Table 9: Training time and model size of energy models



(a) Toxicity Avoidance

(b) Sentiment Control

(c) Formality Transfer

Figure 2: Ground truth label distributions of energy model training data. Notice that Yelp reviews dataset used for training sentiment energy model has labels that are essentially discrete rather than continuous.

Task	Train	Test	Valid
Toxicity Avoidance	47k	3998	5191
Sentiment Control	6,636k	350k	5000
Formality Transfer	4k	-	497

Table 10: Dataset statistics for energy model training

20 and 21 in the Appendix, this pattern of gradient norm-based method yielding improved constraint satisfaction rates with better or comparable fluency and diversity persist in other tasks.

Task	Num. Decoded Tokens	Decoding Time	Computing Infra.
Toxicity Avoidance	55,175	1.2 hr	1 NVIDIA RTX A6000
Sentiment Control (Target sentiment: positive)	22,151	0.32 hr	1 NVIDIA RTX A6000
Sentiment Control (Target sentiment: negative)	22,151	0.50 hr	1 NVIDIA RTX A6000
Formality Transfer (Informal → formal)	18,970	0.47 hr	1 NVIDIA RTX A6000
Formality Transfer (Formal → informal)	14,344	0.70 hr	1 NVIDIA RTX A6000

Table 11: Decoding time of L&E in the main experiments. Although formality transfer tasks are not strictly "decoding" tasks, we refer to the number of tokens in the final style-transferred texts as the number of decoded tokens, and the time it takes to perform the style transfer as the decoding time.

Task	ϵ_{i^*}	(w_{fl}, w_{i^*})	N	m	k	b
Toxicity Avoidance	0.75	(0.1, 0.9)	3	5	10	5
Sentiment Control	0.9	(0.68, 0.32)	10	5	10	3
Formality Transfer	0.9	(0.1, 0.9)	10	5	10	5

Table 12: Hyperparameters used for main experiments. i^* indicates the main constraint, e.g., nontoxicity for toxicity avoidance and positive sentiment for positive sentiment transfer.

	Constraint Sat.	Fluency		Diversity		Content Preserv.	Speed
	→ Inf. ↑	PPL ↓ (Δ)	CoLA ↑	Dist-3 ↑	Rep-3 ↓	$\mathbf{F}_{BERT}$(Base) ↑ (% outputs ≥ 0.5 ↑)	Toks/s ↑
Source Text	0.02	90.79 (0.0)	0.93	0.78	0.00	1.00 (100)	-
Source Text (lower-cased)	0.77	141.63 (50.84)	0.84	0.78	0.00	1.00 (100)	-
Target Text	0.78	364.04 (273.25)	0.76	0.75	0.00	0.63 (83.7)	-
MuCoLa	0.08	61.56 (29.23)	0.81	0.71	0.05	0.63 (63.7)	0.81
Mix&Match	**0.89**	229.43 (138.65)	**0.95**	**0.82**	**0.00**	**0.64 (77.4)**	0.48
Locate&Edit	0.18	**49.68** (41.11)	0.88	0.81	**0.00**	0.50 (46.8)	5.69
L&E *(w/ BERT-base-uncased)*	**0.89**	1033.68 (942.89)	0.70	0.81	**0.00**	0.48 (46.9)	**24.00**

Table 13: Results on formality transfer from formal to informal style.

Method	Reranking Methods	EM Obj.	Toxic Prob. ↓	PPL	CoLA ↑	Dist-3 ↑	REP-3 ↑
Original			0.456	50.923	0.780	0.842	0.000
Locate&Edit	Exhaustive	Reg.	**0.021**	**276.938**	0.767	0.857	**0.000**
		Clsf.	0.028	348.250	**0.794**	**0.859**	**0.000**
	Beam ($f_{fluency} \to \mathcal{E}$)	Reg.	**0.014**	**274.235**	0.760	0.858	**0.000**
		Clsf.	0.031	378.330	**0.780**	**0.864**	**0.000**
	Beam ($\mathcal{E} \to \mathcal{E}$)	Reg.	**0.014**	**274.137**	0.749	0.859	**0.000**
		Clsf.	0.063	784.373	**0.780**	**0.862**	**0.000**
MuCoLa		Reg.	**0.007**	**13.682**	0.700	**0.752**	**0.087**
		Clsf.	0.045	519.215	**0.714**	0.737	0.125
BOLT		Reg.	**0.038**	**10.582**	0.923	0.867	**0.000**
		Clsf.	0.042	10.564	**0.941**	**0.869**	**0.000**

Table 14: Ablation study on the effect of energy model training objective (regression versus binary classification) on toxicity avoidance task.

Method	Reranking Methods	EM Obj.	Positive Prob. ↑	PPL	CoLA ↑	Dist-3 ↑	Rep-3 ↓
Locate&Edit	Exhaustive	Reg.	**0.478**	**72.440**	**0.655**	**0.896**	**0.000**
		Clsf.	0.469	81.896	0.487	0.895	**0.000**
	Beam ($f_{fluency} \to \mathcal{E}$)	Reg.	0.478	59.420	0.469	0.900	**0.000**
		Clsf.	**0.504**	**60.923**	**0.496**	**0.901**	**0.000**
	Beam ($\mathcal{E} \to \mathcal{E}$)	Reg.	0.478	59.184	0.478	0.900	**0.000**
		Clsf.	**0.496**	**59.795**	**0.487**	**0.901**	0.009
MuCoLa	-	Reg.	0.850	8.685	**0.637**	0.620	**0.168**
		Clsf.	**0.903**	**10.020**	0.566	**0.641**	0.195
BOLT	-	Reg.	0.885	**7.589**	**0.973**	**0.848**	**0.000**
		Clsf.	**0.912**	7.363	0.956	0.826	**0.000**

Table 15: Ablation study on the effect of energy model training objective (regression versus binary classification) on sentiment controlled generation task (target sentiment: positive).

Method	**Reranking Methods**	**EM Obj.**	**Formal Prob.** ↑	**PPL**	**CoLA** ↑	**Dist-3** ↑	**Rep-3** ↓
Locate&Edit	Exhaustive	Reg.	0.453	90.189	**0.879**	**0.775**	**0.000**
		Clsf.	**0.458**	**95.133**	0.876	0.775	0.001
	Beam ($f_{fluency} \rightarrow \mathcal{E}$)	Reg.	**0.641**	51.951	**0.818**	0.791	0.007
		Clsf.	0.634	**54.159**	0.795	**0.792**	**0.004**
	Beam ($\mathcal{E} \rightarrow \mathcal{E}$)	Reg.	**0.650**	**47.470**	**0.859**	**0.791**	0.005
		Clsf.	0.636	47.326	0.846	0.790	**0.003**
MuCoLa	-	Reg.	**0.994**	30.680	0.718	**0.786**	0.016
		Clsf.	0.973	**32.690**	**0.722**	0.777	**0.014**
Mix&Match	-	Reg.	**0.033**	551.246	**0.731**	**0.813**	0.001
		Clsf.	0.024	**403.741**	0.683	0.811	**0.000**

Table 16: Ablation study on the effect of energy model training objective (regression versus binary classification) on formality transfer task (informal → formal).

EM Obj.	**Prec.@6**↑	**Rec.@6**↑	**mAP**↑
Reg.	**0.43**	**0.73**	**0.80**
Clsf.	0.38	0.71	0.74

Table 17: Ablation results analyzing the effect of different energy model training objectives on toxic span detection. Across all metrics, using regression-based objective shows better location accuracy than using binary classification objective.

EM Obj.	**Locating Method**	**Prec.@6**↑	**Recall@6**↑	**mAP**↑
Reg.	Grad. Norm	**0.43**	**0.73**	**0.80**
	Attn.	0.39	**0.73**	0.72
Clsf.	Grad. Norm	**0.38**	**0.71**	**0.74**
	Attn.	0.37	0.70	0.70

Table 18: Ablation results on toxic span location using GPT2-L generations data. Gradient norm-based method has better precision, recall, and mAP than attention-based method for both regression-based and classification-based energy models.

Method	**Locating Method**	**Toxic Prob. ↓**	**PPL**	**CoLA ↑**	**Dist-3 ↑**	**Rep-3 ↑**
Exhaustive Search	Grad. Norm	**0.021**	**276.938**	0.767	**0.490**	**0.000**
	Attn.	**0.021**	279.600	**0.777**	0.485	**0.000**
Beam Search	Grad. Norm	**0.014**	**274.137**	0.749	**0.497**	**0.000**
($\mathcal{E} \rightarrow \mathcal{E}$)	Attn.	0.024	277.547	**0.756**	0.492	**0.000**
Beam Search	Grad. Norm	**0.014**	**274.235**	**0.760**	**0.496**	**0.000**
($f_{fluency} \rightarrow \mathcal{E}$)	Attn.	0.024	277.937	0.760	0.491	**0.000**

Table 19: Ablation results on toxicity avoidance comparing different locating method. Across all variations of reranking methods, using gradient norm-based locating results in comparable or superior constraint satisfaction rates.

Method	**Locating Method**	**Positive Prob. ↑**	**PPL**	**CoLA ↑**	**Dist-3 ↑**	**Rep-3 ↑**
Exhaustive Search	Grad. Norm	**0.478**	**72.440**	**0.655**	**0.832**	**0.000**
	Attn.	0.434	81.366	0.566	0.831	**0.000**
Beam Search	Grad. Norm	**0.478**	59.184	0.478	**0.837**	**0.000**
($\mathcal{E} \rightarrow \mathcal{E}$)	Attn.	0.460	**76.124**	**0.504**	0.834	**0.000**
Beam Search	Grad. Norm	**0.478**	59.420	0.469	**0.837**	**0.000**
($f_{fluency} \rightarrow \mathcal{E}$)	Attn.	0.460	**81.301**	**0.487**	0.834	**0.000**

Table 20: Ablation results on sentiment control (target sentiment: positive) comparing different locating method. Across all variations of reranking methods, using gradient norm-based locating results in higher constraint satisfaction rates.

Method	**Locating Method**	**Formal Prob. ↑**	**PPL**	**CoLA ↑**	**Dist-3 ↑**	**Rep-3 ↑**
Exhaustive Search	Grad. Norm	**0.453**	90.189	**0.879**	0.775	**0.000**
	Attn.	0.444	**113.161**	0.834	**0.777**	**0.000**
Beam Search	Grad. Norm	**0.650**	47.470	**0.859**	**0.791**	0.005
($\mathcal{E} \rightarrow \mathcal{E}$)	Attn.	0.456	**111.263**	0.837	0.782	**0.000**
Beam Search	Grad. Norm	**0.641**	51.951	0.818	**0.790**	0.007
($f_{fluency} \rightarrow \mathcal{E}$)	Attn.	0.459	**116.025**	**0.825**	0.782	**0.000**

Table 21: Ablation results on formality transfer (informal → formal) comparing different locating method. Across all variations of reranking methods, using gradient norm-based locating results in higher constraint satisfaction rates.