Mix and Match: Learning-free Controllable Text Generation using Energy Language Models

Anonymous ACL submission

Abstract

Due to the unidirectional nature of prevalent autoregressive generation models, recent work on controlled generation based on global text attributes has either required attribute-based fine-tuning of the base language model, or restricted the parametrization of the attribute prediction model to be compatible with the base LM. In this work, we propose Mix and Match LM, a global score-based alternative for controllable text generation that combines arbitrary pretrained black-box models for achieving the desired attributes in the generated text without involving any fine-tuning or structural assumptions about the blackbox models. We interpret the task of controllable generation as drawing samples from an energy-based model whose energy values are a linear combination of scores from blackbox models that are separately responsible for fluency, the control attribute, and faithfulness to any conditioning context. We use a Metropolis Hastings sampling scheme to sample from this energy-based model using bidirectional context and global attribute features. We validate the effectiveness of our approach on various controlled generation and style-based text revision tasks by outperforming recently proposed methods that involve extra training, fine-tuning, or restrictive assumptions over the form of models.

1 Introduction

While large transformer-based autoregressive language models trained on massive amounts of data found on the internet exhibit exceptional capabilities to generate natural language text, effective methods for generating text that satisfy global constraints and possess holistic desired attributes remains an active area of research. These mechanisms for controlling the generation of language have potential to mitigate undesirable biases encoded by the large language models and prevent generation of hate speech and toxic language (Yang and Klein, 2021; Xu et al.; Gehman et al., 2020). Much of the prior work has approached controlled generation via either training (Prabhumoye et al., 2020; He et al., 2020; Lample et al., 2018; Shen et al., 2017; Krishna et al., 2020; Reif et al., 2021; Ficler and Goldberg, 2017; Khalifa et al., 2021) domain-conditioned neural language models or finetuning/modifying an underlying large pretrained base model for generation on domain-specific data (Ziegler et al., 2019; Keskar et al., 2019; Mai et al., 2020; Gururangan et al., 2020; Krause et al., 2020) for attribute sensitive generation. Not only do these approaches involve computational overhead and estimation errors associated with training of language models, crucially they are also dependent on access to a large amount of attribute-specific language data which can be impractical in many scenarios and exacerbate privacy concerns.

Our approach eschews training and focuses on generation-time control from pretrained modules. Recent work in this space has used attribute discriminators (Dathathri et al., 2020; Krause et al., 2020; Yang and Klein, 2021; Holtzman et al., 2018) to steer the generation from a large autoregressive language model. These discriminators need to be separately trained on partial generations in order to be operationalized with step-wise autoregressive models. As a result, this approach also requires availability of data to train step-wise discriminators for attributes that are essentially global (at the sequence-level) in nature. Therefore, we focus on drawing samples from a test-time combination of *pretrained blackbox* experts that each score a desired property of output text – for example, fluency, attribute sensitivity, or faithfulness to the context. Specifically, we view the product of these blackbox experts as a probabilistic energy model (Hinton, 2002) – i.e., a *non-autoregressive*, globally normalized language model – and then sample (without further training or fine-tuning) using a specialized Gibbs sampler with a Metropolis-Hastings correction step (Goyal et al., 2021).

Our full framework, which we entitle Mix and Match LM (depicted in Figure 1), enables generation of high-quality attribute-controlled samples by
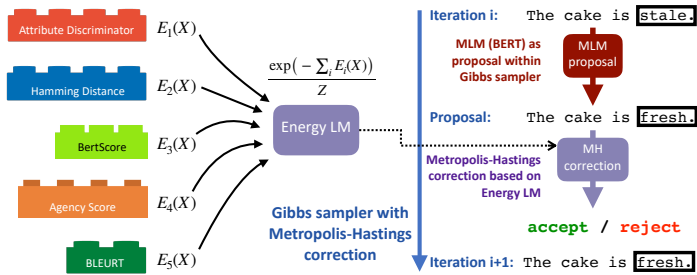
Figure 1: Overview of Mix and Match LM. The Lego pieces show different experts that can be used to form the energy LM and help control different features in the generated text. The right side shows the ith step in the the Gibbs sampling chain, where a proposal is made by the MLM, and then it is accepted/rejected based on the energy score.

mixing and matching blackbox models like off-the-shelf pretrained attribute-sensitive discriminators (e.g., sentiment classifiers), large bidirectional pretrained language models like BERT (Devlin et al., 2019), and other modules specializing in capturing desirable features pertaining to faithfulness to any additional context, like hamming distance, or BertScore distance (Zhang et al., 2020) between the sample and the conditioning context. We generate samples from the energy language model assembled from these component experts by using the recently proposed Gibbs-Metropolis-Hastings scheme (Goyal et al., 2021) for sampling from energy models using a masked language model as a proposal distribution. In this scheme, an expressive bidirectional language model like BERT is used to make a proposal at each transition step in the Gibbs chain to jump to a sequence $\bar{\mathbf{x}}$ from the current sequence $\mathbf{x}$. This proposal's fitness is judged by the change in the energy language model's score, with the sampler accepting proposals with larger energy reductions at a higher rate. While the MCMC nature of our sampler negatively impacts the runtime during decoding compared to autoregressive approaches with ancestral sampling, we find our approach to still be practical and yielding high-quality diverse samples that respect the distribution induced by the product of expert blackbox models.

We demonstrate the flexibility of our approach by performing a variety of controlled generation tasks, such as aspect-based text revision, style transfer, and attribute grounded generation and compare it to recently proposed controlled generation approaches that are more resource/data intensive. We observe that our approach, which does not require any gradient optimization and is able to combine arbitrary heterogeneous blackbox models, outperforms other approaches according to various automated metrics of fluency, quality, and control, as well as human evaluations. We have provided code, data and sample generations as supplementary attachments to our submission (see A.1).

Related Work: The approaches closest in spirit to our work involve steering generation from a base language model with external attribute-sensitive control mechanisms. Plug-and-Play LM (Dathathri et al., 2020) uses discriminators learned from an autoregressive LM's top-level hidden layer to modify the LM's states toward increasing probability of the desired attribute via gradient ascent at each step. GeDi (Krause et al., 2020) and FUDGE (Yang and Klein, 2021) take a similar approach but train custom step-wise attribute-sensitive discriminators that decide whether the desired attribute is likely to be satisfied by the current generation path. GeDi trains class-conditional language models for these discriminators and hence additionally relies on access to attribute sensitive language data. DExperts (Liu et al., 2021) is another decoding-time controllable generation approach which modifies the step-wise softmax logits of an autoregressive pretrained LM with softmax logits of separately trained domain-specific *expert* autoregressive language models. These approaches require training of custom modules and do not enjoy the benefits of incorporating global attribute-based features into the generation mechanism in a simple probabilistic manner. In contrast, our energy-based formulation is not only optimization-free, but also fully modular and able to easily incorporate global features, allowing for heterogeneous blackbox experts to be combined with each other.

2 Mix-and-match Language Models

In this section, we describe our approach and motivation behind our method. Specifically, we frame the problem of performing controlled generation as a problem of sampling from a specialized energy-based (or globally normalized) sequence model that defines a probability distribution which satisfies the desired constraints we wish to impose in the controlled generation setting. As described below, this energy based model is composed of pretrained components and does not require any further optimization. An energy-based sequence model defines the probability distribution over the space of pos-

2

sible sequences $\mathcal{X}$ as:[1] $p(X;\theta) = \frac{e^{-E(X;\theta)}}{\sum_{X' \in \mathcal{X}} e^{-E(X';\theta)}}$, where $E(X;\theta)$ refers to the scalar energy of a sequence X that is parametrized by θ. Lower energy corresponds to higher likelihood of X. In contrast to the common autoregressive sequence models, exact likelihood computation and efficient sampling from these models is challenging. Despite these challenges, we focus on this paradigm of sequence modeling because energy-based models offer increased flexibility via sequence level features and constraints. As we discuss next, this capability lets us easily define expressive functions for controlled generation of sequences which is not readily offered by the autoregressive modeling paradigm.

2.1 Product of Experts Energy-based Models and Controlled Generation

Our approach is motivated by the perspective that the task of controlled generation requires concentrating probability mass over small subspace of sequences in $\mathcal{X}$ that satisfies various constraints pertaining to fluency, target attributes, and other control variables. Consider the task of generating positive sentiment sentences. This requires satisfaction of two major constraints: (1) The sequence X should be well-formed, (2) The sequence X should express positive sentiment. If we have access to two separate probability distributions over $\mathcal{X}$, one for modelling well-formedness ($p_1(X)$) and another for modelling positivity ($p_2(X)$), then a natural solution for controlled generation in this setting would be to draw samples from a probability distribution that is a product of these two distributions i.e. $p_{\text{desire}}(X) \propto p_1(X) \cdot p_2(X)$. In our approach, we further relax this requirement by assuming access to *expert blackboxes* that yield scalar non-probabilistic energy scores E_1 and E_2 indicating fitness of a sequence w.r.t. well-formedness and positivity respectively. Under the product of experts framework above the desired probability distribution would take the form: $\log p_{\text{desire}}(X) = -(E_1(X) + E_2(X)) - \log Z$. This expression shows that when working with scalar scores for the expert blackboxes, the product of expert models yields an energy model whose energy is simply the sum of the scalar energy values obtained from the expert models. Inspired by this, we propose a framework for controlled generation that involves linear combinations of various blackbox experts in order to obtain a distribution whose samples satisfy the requirements of a desired con-

trolled generation task: $E_{\text{M\&M}}(X) = \sum_{i=1}^{k} \alpha_i E_i(X)$, where our proposed *mix-and-match* energy is composed of k expert energy components, which are weighted by scalar hyperparameters α.

2.2 Expert Factors in Mix-and-Match LM

As shown in Fig. 1, we use the following blackbox experts in our experiments as modules that we can add or remove to produce desired behavior:

$\mathbf{E}_{\text{mlm}}(\mathbf{X})$: Recent work has shown that large masked language models (MLM) like BERT can discriminate between well-formed and ill-formed sentences (Zhang et al., 2020) and induce an implicit energy function over the sequences (Goyal et al., 2021). Hence, we use BERT-base as a blackbox to model the form and fluency of sentences. Specifically, we use an energy parametrization introduced in Goyal et al. (2021) which is negative of the sum of unnormalized logits at each position obtained via forward pass of the MLM after masking the respective positions iteratively.

$\mathbf{E}_{\text{disc}}(\mathbf{X})$: This particular expert module refers to the energy obtained via the discriminator for the attributes of interest. What this module returns is the raw logits of the discriminator, for the target attribute. For instance, if we have a sentiment classifier, and want to produce positive sentiment, then $E_{\text{disc}}(X) = -\log p(+|X)$.

$\mathbf{E}_{\text{hamm}}(\mathbf{X};\mathbf{X}')$: For a given sequence X', this quantity refers to the hamming distance between the sequence X and X'. This penalizes token level deviation from X' which is useful if we are interested in only making minor edits to X' as described later.

$\mathbf{E}_{\text{fuzzy}}(\mathbf{X};\mathbf{X}')$: Similar to the hamming distance, this quantity refers to the BertScore (Zhang et al., 2020) computed between X and X' which can be viewed as a *fuzzy* hamming distance that takes semantic similarity into account.

2.3 Sampling scheme

To sample from the energy parametrizations described in the previous section, we follow the Metroplolis Hastings (Hastings, 1970) MCMC scheme for sampling from the masked language models introduced by Goyal et al. (2021). While the proposal distribution we use is the same as Goyal et al. (2021) i.e. masked language model's (BERT's) conditionals, the energy parametrizations we use are more suitably designed for controlled generation.

We briefly explain the sampling procedure, which involves forming long Markov chains of sequences starting with a random sequence, and following the MH scheme which uses a proposal

---

[1] For simplicity, we are concerned with a finite set of sequences limited by some maximum length.

distribution to propose a new sequence at each step in a chain which is either accepted or rejected based on its fitness to the energy function. The sequences at the end of these chains correspond to samples from the desired energy-based model. Operationally, at each MCMC step, we mask out a token at a random position in the current sequence X in the chain, and propose a new sequence $\bar{X}$ to transition to by sampling a token from the MLM conditional softmax at the masked position. This proposed sequence is evaluated by its ability to reduce the energy from the current sequence in the chain and is accepted with the probability $p(\bar{X}; X) = \min\left(1, \frac{e^{-E_{\text{M\&M}}(\bar{X})} \, p_{\text{mlm}}(X_i|X_{\setminus i})}{e^{-E_{\text{M\&M}}(X)} \, p_{\text{mlm}}(\bar{X}_i|X_{\setminus i})}\right)$. $E_{M\&M}(X)$ refers to the product of experts energy, i refers to the position chosen for masking, p_{mlm} refers to the MLM's conditional distribution at the [MASK] position. Intuitively, this acceptance probability indicates that the proposed sequence $\bar{X}$ is more acceptable if it has lower energy than the current sequence X in the chain and is rare or less likely to be proposed by the proposal distribution again.

2.4 Controlled generation Tasks

We use the expert blackbox factors and the sampling scheme describe above in our framework to perform two kinds of controlled generation tasks.

Prompted generation: This task focuses on generating well-formed sentences that start with a specified prompt and also satisfy a target attribute for which we have access to a discriminator. An example task would be to generate positive sentiment sequences starting with `This movie`. The energy function takes the form:

$$E_{\text{gen}}(X) = E_{\text{mlm}}(X) + \alpha \, E_{\text{disc}}(X) \quad (1)$$

α is a hyperparameter that controls the tradeoff between the MLM score and the discriminator's influence. For MH-based sampling for this task, we initialize the sequence with the starting prompt and rest of the tokens masked out, which creates a seed text of shape `the movie [MASK] [MASK] ... [MASK]`, for the prompt example of `the movie`. The number of mask tokens depends on the target generation length, and we constrain the sampler to only produce proposals and revise non-prompt tokens, and mark the prompt tokens as "frozen".

Controlled text revision: This task involves editing a source sequence X' in order to satisfy the desired target attributes exhibited by the generated sequence X. The energy function for this task is:

$$E_{\text{rev}}(X) = E_{\text{gen}}(X) + \beta \, E_{\text{hamm}}(X, X') + \gamma \, E_{\text{fuzzy}}(X, X') \quad (2)$$

This energy function in addition to valuing well-formedness and satisfying target attribute requirements, also focuses on maintaining faithfulness to the source sequence X'. For sampling with this energy, we initialize the sequence with the sequence X' to be edited. This sets the length of the target sequence to be the same as the source. In this setup, the sampler can revise all tokens and is not constrained.

For both these tasks, we run a separate MCMC chain for each generated sentence for 8 to 15 epochs, depending on the task. An epoch refers to one masking cycle over all the non-frozen positions (selected randomly) of the sequence.

3 Experimental Setup

We provide full experimental details in appendix Section B, here we provide a brief overview of the tasks, datasets, baselines and metrics used in the experiments.

3.1 Tasks and Datasets

Controllable debiasing (ROC story corpus): We use the subset of the ROC story corpus (Mostafazadeh et al., 2016) test-set that is used by PowerTransformer (Ma et al., 2020) for their evaluations. We use this data for controllable debiasing, a text revision task which aims to correct the implicit and potentially undesirable agency biases in character portrayals, by replacing verbs such as wish and dream, with pursue and achieve.

Sentiment transfer (Yelp): We use Yelp (Shen et al., 2017) dataset's test-set for the task of sentiment transfer. The test set comprises of 1000 sentences, half with positive and half with negative sentiment. We also have a reference set of hand written sentiment transferred sentences, provided by (He et al., 2020) that we use for reporting evaluation metrics.

Formality transfer (GYAFC): We use 1051 sentences from the entertainment and music domain subset of the GYAFC (Rao and Tetreault, 2018) dataset, which contains formal and informal sentences for the task of formality transfer (both directions of formal to informal and informal to formal).

Prompted generation: We evaluate our approach on two forms of prompted generation: 1) sentiment controlled generation, and 2) topic controlled generation. For sentiment controlled generation, we set Mix and Match LM to generate text with

positive or negative sentiment given prompts, by using a Yelp sentiment classifier as discriminator and compare against PPLM (Dathathri et al., 2020) which is a popular sentiment controlled generation method. For topic controlled generation, we compare against FUDGE (Yang and Klein, 2021), and follow their experimental setup consisting of 7 distinct topics and 20 prompts.

3.2 Expert Component Configurations

We use a Huggingface pre-trained `bert-base-uncased` model as our MLM for yielding $E_{\texttt{mlm}}$ and also providing the proposal distribution in our MH MCMC sampler. For obtaining $E_{\texttt{disc}}$, we train BERT-based classifiers on the training-set of our datasets to use as our attribute discriminators. We could have used any pre-trained attribute classifier from Huggingface for $E_{\texttt{disc}}$, but we keep those aside to use as external attribute classifiers for fair evaluation against baselines. For experiments in which we add the BertScore (Zhang et al., 2020) component to the energy, we use the pre-trained `roberta-large_L17` model.

3.3 Baselines

PowerTransformer. For the task of controllable debiasing (agency revision), we compare our work with PowerTransformer (Ma et al., 2020), an approach that uses paraphrasing and self-supervision based on a reconstruction loss, building on pre-trained language models, to re-write text and control agency level of sentences.

He et al. For style transfer on sentiment an formality, we compare with He et al. (2020), a generative style transfer framework which uses a variational autoencoder (VAE) built using a sequence-to-sequence LSTM-based model to do unsupervised style transfer. This framework needs to be trained from scratch for each style transfer task.

UNMT. As a second baseline for style transfer, we use UNMT (Lample et al., 2018), an unsupervised machine translation framework that demonstrates high performance for sentiment transfer.

PPLM. For the task of sentiment controlled generation, we compare to Plug-and-Play LM (PPLM) Dathathri et al. (2020), which does attribute controlled generation using the flow of gradients from discriminators trained on the last hidden layer representations of the generator, to guide generation.

FUDGE. This approach (Yang and Klein, 2021) trains step-wise discriminators on partial generations from GPT-2 to determine whether the

constraints related to desired attributes will be satisfied by the future completion of the sequence or not. We compare against this on topic controlled generation as this approach was shown to be superior to PPLM on this task.

3.4 Evaluation Metrics

We use a variety of evaluation metrics to compare our approach's performance on two major facets: (1) Quality of generated text, and (2) success on matching the target attribute used for control.

3.4.1 Text Quality and Semantic Similarity

GPT-2 PPL. We feed our generated test sentences to a Huggingface (Radford et al., 2019) pre-trained GPT-2 xl model, and report its perplexity (PPL), as an automatic measure of fluency. Although this measure is not a perfect indicator of fluency, we find it to be a useful metric alongside human judgements. [2]

BLEU. For sentiment (Yelp) and formality (GYAFC) transfer where we have reference text, we report the BLEU score. For controlled debiasing, we report BLEU between generated text and source, and show it as BLEU (src).

BertScore. As a measure of meaning preservation, we use the F1 BertScore metric (Zhang et al., 2020) to compare the semantic similarity of the provided reference sentence with the generated output.

Hamming Distance. We also report the hamming distance between the source text and generated text, to measure the extent of the change.

3.4.2 Attribute Quality

Internal Classifier Accuracy. We report accuracy of the internal classifier (the discriminator used for generation) on the generated text, assuming the target attribute is the correct label. The higher this accuracy is, the better.

External Classifier Accuracy. It is natural to get high accuracy on the internal classifier, since we are sampling from it. To have a fair comparison, we report accuracy using external classifiers from Huggingface (`textattack/bert-base-uncased-yelp-polarity` (Morris et al., 2020) for sentiment and `cointegrated/roberta-base-formality` for formality).

Agency Lexicon Accuracy. For controlled debiasing, we measure the accuracy of the change in agency by comparing the target agency level

---

[2]Due to the high variance in the PPL scores generated across sentences by GPT-2, we report the median score for each system under comparison.

with that of the generated text, extracted using the connotation frames lexicon, and following the setup from Ma et al. (2020).

4 Results

4.1 Controllable Debiasing

Table 1 shows our results for the task of text revision for controlling agency bias which is introduced by PowerTransformer Ma et al. 2020, our Baseline for this task. PowerTransformer has a vanilla (no boost) variant and a variant with vocab boosting, which up-weights the logits of verbs that belong to the target agency lexicon so as to increase their probability and incentivize generation in that direction. We also measure our metrics on the original test-set, without revision, to provide a better sense of the changes made.

We offer different variants of our framework, to provide a fair comparison and to better ablate our proposed method. "Disc" denotes our framework where we add the discriminator expert ($E_{\texttt{disc}}$) which is trained to predict the agency level of a sentence, to the energy along with $E_{\texttt{mlm}}$, and $E_{\texttt{hamm}}$ (Eq. 2). Hamming distance is computed between the generated proposals and the source sentence. The "Agency Score" variant adds an alternative term to $E_{\texttt{M\&M}}$ instead of $E_{\texttt{disc}}$, which is the number of target agency verbs according to the connotation frames lexicon (Sap et al., 2017) in the sentence. The "Disc+Agency" variant has both the energy components. We also apply our method in two ways: "Verb Replace" which allows the sampler to propose revisions for only one pre-determined verb (provided in the dataset). In this setup all tokens remain frozen, except for the given verb. The conventional mode (M&M LM), however, proposes revisions for all tokens in the sentence and is not constrained.

Table 1 shows that in the conventional setup, Mix and Match LM (Disc only) has performance similar to that of PowerTransformer, without boosting. With the Agency Score component, our method outperforms PowerTransformer in terms of accuracy of revision as per the agency lexicon accuracy metric, with negligible loss in meaning (BertScore). The reason behind this better performance in terms of applying target agency accuracy is that our method's sampling is guided by the energy that is directly built on the metrics we care about, as opposed to trying to apply them through paraphrasing and proxies such as vocab boosting, which are employed in the PowerTransformer method.

Another important observation here is the dif-

ference between "Verb Replace" and conventional modes. This ablation shows that although our method makes few changes (the average hamming distance between source and output sentences are between 1.37 and 2.45), it still outperforms a "static" method that has extra knowledge of the offending verb and focuses on changing only that verb, by a significant margin.

4.2 Style Transfer

In this section we experiment with sentiment and formality transfer, where Sentiment transfer needs fewer changes and formality transfer needs more structural change to the original sentence.

4.2.1 Sentiment Transfer

For this task we include two components in our energy model, the attribute discriminator ($E_{\texttt{disc}}$), to induce the target style, and the hamming distance ($E_{\texttt{disc}}$), to maintain the meaning of the sentence. We don't include the more complex semantic similarity-related component like $E_{\texttt{fuzzy}}$, since sentiment transfer can normally be done by making only a few changes to the sentence. We report results with two different variants, one where the discriminator component has a higher coefficient in the energy (Discriminator↑) and one where the hamming distance has a higher coefficient (Hamming↑). In effect, these two show the trade-off between transfer quality and language quality.

We see in Table 2 that our method, with the hamming component up-weighted, outperforms both the generative baselines in terms of transfer accuracy (Ext. Clsf.) and semantic similarity (BertScore). We can also see Mix and Match LM has higher BLEU score, with respect to the provided handwritten reference sentences. We hypothesize that this superiority is due to the tendency of our model to make minimal revisions that satisfy the product of experts energy model. Therefore, our model can successfully change the style without changing the meaning of the sentence. The generative baselines however, regenerate the sentence which imposes more change, as can be observed from the hamming distance column (Hamm.(src)) in Table 2.

4.2.2 Formality Transfer

For this task, we include the formality classifier ($E_{\texttt{disc}}$), Hamming distance ($E_{\texttt{hamm}}$), and BertScore ($E_{\texttt{fuzzy}}$) components in the energy formulation, to permit the transfer of style and also maintain the meaning of the sentence. $E_{\texttt{fuzzy}}$ helps with imposing semantic similarity between source

Table 1: Controllable debiasing/ sentence agency revision on ROC-story corpus. The *(src)* next to the metrics denotes measurement with respect to the source text. *Int. Clsf.* is the accuracy of the discriminator used in the energy. *Hamm.* shows the Hamming distance. *Agency Acc.* is the accuracy of agency revision based on the agency lexicon (Sec B.4.1).

	Method	BLEU(src)	GPT-2	BertScore(src)	Hamm.(src)	Int. Clsf.	Agency Acc.
	Source Text	100.00	153.9	1.00	0.00	7.47	9.81
Basel.	**PowerTransformer (No Boost)**	60.30	210.8	**0.94**	1.11	64.84	**69.17**
	PowerTransformer (+Boost)	57.46	247.2	**0.95**	1.28	77.23	**85.03**
Ours	M&M LM Verb Replace (Disc)	60.53	238.7	0.95	1.04	81.05	70.80
	M&M LM Verb Replace (Agency Score)	51.95	193.3	0.96	0.89	32.42	64.75
	M&M LM Verb Replace (Disc+Agency Score)	54.52	248.8	0.95	1.05	77.23	77.27
	M&M LM (Hamming +Disc)	56.26	211.2	**0.95**	1.37	96.52	**69.00**
	M&M LM (Hamming+Agency Score)	51.95	231.6	0.95	1.56	23.13	86.01
	M&M LM (Hamming+Disc+Agency score)	39.82	261.6	**0.93**	2.45	90.16	**89.42**

Table 2: Sentiment transfer on Yelp. *(ref)/(src)* means the metric measured is measured with respect to reference/source text. *Int./Ext. Clsf.* show internal/external attribute classifier accuracy. *Hamm.* shows Hamming distance.

	Method	BLEU(ref)	GPT-2	BertScore(src)	Hamm.(src)	Int. Clsf.	Ext. Clsf.
	Reference Text	100.00	169.5	1.00	5.80	83.70	85.60
Basel.	He et al.	18.67	200.6	0.93	4.23	84.87	79.82
	UNMT	17.00	171.8	**0.94**	3.67	84.87	**80.22**
Ours	M&M LM (Discriminator ↑)	15.75	163.5	0.93	2.84	97.53	90.00
	M&M LM (Hamming↑)	19.71	191.5	**0.95**	1.83	94.72	**82.85**

Table 3: Formality transfer on GYAFC dataset. The *(ref)/(src)* next to the metrics denotes that they are measured with respect to the reference/source text. *Int. Clsf.* shows the accuracy of the discriminator used in the energy, and *→Informal/Form.* shows the breakdown of the external classifier accuracy. *Hamm.* shows the Hamming distance.

	Method	BLEU(ref)	GPT-2	BertScore(src)	Hamm.(src)	Int. Clsf.	→Informal	→Form.
	Reference Text	100.00	118.1	0.92	7.72	82.97	100.00	9.41
Basel.	He et al.	15.83	122.8	**0.90**	10.03	64.79	**100.00**	3.33
	UNMT	14.17	143.8	0.90	11.92	56.04	99.81	7.64
Ours	M&M LM (Discriminator ↑)	17.78	206.3	0.89	5.22	91.15	96.67	23.13
	M&M LM (BertScore↑)	27.71	194.4	**0.93**	2.50	72.12	**94.26**	19.01

and generated sentences, since Hamming alone isn't sufficient for judging comparable formal and informal sentences. We show results for two setups of our framework, one where the discriminator coefficient is higher (Discriminator↑) and another where the BertScore coefficient is higher (BertScore↑).

In Table 3 we have broken down the external classifier accuracy for the different transfer directions of formal to informal (→ Inf.) and vice versa. We do this because the → Form. task is generally harder and therefore has lower accuracy. We observe that our method outperforms the baselines in terms of BertScore and BLEU, for similar levels of external classifier accuracy. However, we can see that the GPT-2 PPL of our method is higher the baselines. The reason behind this is the format and noise in the data. The samples for this dataset are taken from the music and entertainment industry domain, and contain some symbols and characters similar to emojies (e.g. ":)" and "***"). This is where the tendency of our approach toward minimal revisions is hurtful–

our revisions of text, often do not get rid of all of these symbols, while the baselines' generative methods successfully remove all the superfluous characters because they rewrite sentences from scratch.

4.3 Prompted Controlled Generation

4.3.1 Sentiment Controlled Generation

We generate 560 sequences of different lengths (12, 20 and 50 tokens), given 14 prompts, 2 sentiments and 20 sequences per sentiment, taken from Dathathri et al. (2020)'s experimental setup. The prompts and sample generations are in the appendix B.9 and A.2, and full list of generations is in the supplementary material.

Table 4 shows our results for this experiment. Here, we have an additional metric, the MLM energy (lower is better), which, like GPT-2, indicates the quality of generated sentences (Salazar et al., 2020) according to BERT. We report this extra metric here since PPLM uses a GPT model for generation, and it is natural that it would measure better on this metric.

Table 4: Prompted sentiment controlled generation results and human evaluations. *BERT* denotes the BERT MLM energy score (equivalent of GPT-2 perplexity), and lower score is better. *Int./Ext. Clsf.* show the accuracy of the discriminator used in the energy/external discriminator from Huggingface.

Length	GPT-2 ($\downarrow$)		BERT ($\downarrow$)		Int. Clsf. ($\uparrow$)		Ext. Clsf. ($\uparrow$)		Human Preference (%)	
	Ours	PPLM	Ours	PPLM	Ours	PPLM	Ours	PPLM	Ours	PPLM
12	264.1	113.1	-160.4	-137.1	94.3	71.7	65.1	58.0	71.1	29.9
20	167.2	61.1	-271.0	-237.1	96.3	74.5	65.9	57.6	62.9	37.1
50	122.3	29.0	-692.3	-606.1	93.8	73.6	68.6	60.7	46.7	53.3

The table shows that for all lengths of generated sentences, our method is much better at inducing the target sentiment. However, in terms of GPT-2 PPL and MLM score, PPLM and Mix and Match performs better, respectively, as the former uses GPT-2 for generation and the latter uses BERT. To enable a more conclusive comparison of the text quality, we report results with human evaluations. For these evaluations, we randomly select 10 generated outputs for each prompt, per sentiment (240 overall), and asked three Amazon Turkers per sample pair, which sample they find more fluent. We report the majority vote of the Turkers in the table. The results show that for sequences with lengths 12 and 20, they found our generations more fluent. However, for length 50, the preference rate for M&M drops to 46.7%, which shows that our method is superior to PPLM for short/medium length generation, however PPLM does better at generating longer sequences.

4.3.2 Topic Controlled Generation

We follow FUDGE's (Yang and Klein, 2021) experimental setup which covers 7 topics, given 20 prompts and generate 7×20 sequences of length 20. To enforce topicality on our generations, we add a topic-based energy, E_{topic}. This energy is essentially the negative count of the number of topic-related words (using the list provided by FUDGE).

Table 5 shows the results of this experiment, generations are also provided in A.2. Topic-score ($\uparrow$) is the usage rate of topic related words that were used for training and evaluation of topic controlled generation by Yang and Klein in their paper. Grammaticality ($\uparrow$) is the probability of grammaticality given by a Roberta-based CoLA grammaticality model averaged over all outputs (Warstadt et al., 2019). The "Div" ($\uparrow$) metrics show diversity of generated text, over unigrams, bigrams and trigrams. Finally, the human evaluations show human preference, in terms of fluency of the sentences (B.10). As shown by the table, the fluency of our method is comparable to that of FUDGE, even better in terms of human preference and grammaticality judgement.

Table 5: Prompted topic controlled generation results and human evaluations.

Metrics	FUDGE	**M&M LM**
Topic-score ($\uparrow$)	1.45	1.21
Grammaticality ($\uparrow$)	0.61	0.74
GPT-2 PPL ($\downarrow$)	104.8	110.2
Diversity over Unigrams ($\uparrow$)	0.54	0.57
Diversity over Bigrams ($\uparrow$)	0.86	0.89
Diversity over Trigrams ($\uparrow$)	0.87	0.88
Human Preference(%) ($\uparrow$)	37.5	62.5

FUDGE has a slightly higher topic-score, which is expected, since it trains a custom step-wise discriminator for each topic that is optimized for the task. But our approach shows competitive faithfulness to the topics especially considering the fact that prompted GPT-2 generations without the FUDGE discriminators only achieve a topic-score of 0.23.

4.4 Inference Speed

Given that our model's inference procedure involves MCMC sampling, its reasonable to expect its runtime to be slower than more traditional baselines. We find that our un-optimized implementation requires 8 seconds per generation and 3 seconds per revision (for sequences of length 20), while in contrast baseline system PPLM requires 16 seconds and FUDGE requires 0.4 seconds. This is a substantial slowdown, but not one that renders the proposed approach impractical in offline settings. Further, faster sampling schemes are beyond the scope of this paper, but might be explored in future work to speed up models like M&M LM.

5 Conclusion

We present Mix and Match Language Models (M&M LMs), a training-free framework for controlled text generation that can easily mix heterogeneous expert modules. We show that our framework outperforms prior methods on a suite of text revision and attribute controlled generation tasks. Further, our results indicate that probabilistic energy language models, typically considered intractable, can be used for practical text generation tasks when combined with an appropriate sampling scheme.

Ethical Considerations

We have designed our framework with re-usability and modularity in mind, so as to alleviate the need of multiple training and fine-tuning rounds, and to reduce the negative environmental effects that training large models have. We do however acknowledge that strong controlled generation methods that rely on discriminators can have the potential to regurgitate the training data and produce harmful outputs and toxic language (Xu et al.; Gehman et al., 2020; Wallace et al., 2020). However, if used properly and for good, we anticipate positive impact on debiasing and safe generation.

References

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. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2021. Exposing the implicit energy networks behind masked language models via metropolis-hastings. *ArXiv*, abs/2106.02736.

Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark. 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.

W Keith Hastings. 1970. Monte carlo sampling methods using markov chains and their applications.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. A distributional approach to controlled text generation. In *International Conference on Learning Representations*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. GeDi: Generative Discriminator Guided Sequence Generation. *arXiv preprint arXiv:2009.06367*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *ArXiv*, abs/2010.05700.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

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.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. 2020. PowerTransformer: Unsupervised controllable revision for biased language correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7426–7441, Online. Association for Computational Linguistics.

Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, and James Henderson. 2020. Plug and play autoencoders for conditional text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6076–6092, Online. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *NAACL*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Maarten Sap, Marcella Cindy Prasettio, Ari Holtzman, Hannah Rashkin, and Yejin Choi. 2017. Connotation frames of power and agency in modern films. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2329–2334, Copenhagen, Denmark. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6833–6844.

Eric Wallace, Mitchell Stern, and Dawn Xiaodong Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *EMNLP*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, Dan Klein, and UC Berkeley. Detoxifying language models risks marginalizing minority voices.

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. In *International Conference on Learning Representations*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Appendix

A.1 Code and Data Directory Structure

We have provided all our code (under software), data and our generations (under data) through the submission platform, and our checkpoints are uploaded anonymously here https://zenodo.org/record/5855005. There is a readme file inside the code, which has instructions on how to run generation and get evaluation metrics.

Here is a short guide on how to navigate the files: Under software, you see `mix_match_-code`, which contains all the scripts for running generation and evaluation. Under data, you would see one compressed file, comprising of two directories, `mix_match_generations` and `mix_match_data`. The first one has the sample generations, and contains two folders, one for human evaluations against FUDGE and PPLM (`human_evals_fudg_pplm`), and the other for automatic evaluations, for sentiment and bias (`output_samples_bias_sentiment`). We have not included the data files for the formality, since the GYAFC dataset requires permission for access, so we cannot release it.

The `mix_match_data/clsf_data` folder contains training samples for training the classifiers, which could be avoided if huggingface classifiers are used. `mix_match_data/data` contains the prompts/original sentences used for generation/transfer.

A.2 Sample Generations

Due to page limitations in the body of the paper, we include more sample generations from our method in the form of tables here (all of them are attached as supplementary material to the manuscript). We have no samples from the formality transfer task, however, since the data used (GYAFC) is protected and needs permissions for access, so we cannot publish it. However, we have provided code needed to reproduce our results, once access to the original data is gained.

Table 8 shows some original and transferred samples, for the de-baising and sentiment transfer tasks. Tables 7 and 6 show sample PPLM and FUDGE generations versus Mix and Match generations.

B Experimental Setup Details

B.1 Tasks and Datasets

Controllable debiasing (ROC story corpus): We use the subset of the ROC story corpus (Mostafazadeh et al., 2016) test-set that is used by PowerTransformer (Ma et al., 2020) for their evaluations. We use this data for controllable debiasing, a text revision task which aims to correct the implicit and potentially undesirable agency biases in character portrayals. This test-set consists of 549 sentences, where 224 sentences have low agency verbs (such as wish, dream, etc.) and the rest have high agency (like pursue, achieve, etc.). The task is to revise the sentences such that the meaning is preserved, but the agency of the sentence is changed in the target direction.

Sentiment transfer (Yelp): We use Yelp (Shen et al., 2017) dataset's test-set for the task of sentiment transfer. The test set comprises of 1000 sentences, half with positive and half with negative sentiment. We also have a reference set of hand written sentiment transferred sentences, provided by (He et al., 2020) that we use for reporting evaluation metrics.

Formality transfer (GYAFC): We use 1051 sentences from the test-set of the GYAFC (Rao and Tetreault, 2018) dataset, which contains formal and informal sentences for the task of formality transfer (both directions of formal to informal and informal to formal). Here we use the entertainment and music domain subset of this data, following the evaluation setup of (He et al., 2020). This dataset also contains parallel data between formal and informal sentences, which we use as reference for reporting evaluation metrics.

Prompted generation: We evaluate our approach on two forms of prompted generation: 1) sentiment controlled generation, and 2) topic controlled generation. on prompted generation. For sentiment controlled generation, we set Mix and Match LM to generate text with positive or negative sentiment given prompts (listed in Appendix B.9) by using a Yelp sentiment classifier as discriminator and compare against PPLM (Dathathri et al., 2020) which is a popular sentiment controlled generation method. For topic controlled generation, we compare against FUDGE (Yang and Klein, 2021), and follow their experimental setup consisting of 7 distinct topics and 20 prompts.

B.2 Expert Component Configurations

We use a Huggingface pre-trained `bert-base-uncased` model as our MLM for yielding $E_{\texttt{mlm}}$ and also providing the proposal distribution in our MH MCMC sampler. For obtaining $E_{\texttt{disc}}$, we train BERT-based classifiers on the training-set of our datasets to use as our attribute discriminators. Al-

Table 6: Original vs. style transferred sample sentences, using Mix & Match LM.

	Original	Transferred
De-biasing	whitney is going to fail her test .	whitney is set to get her test .
	mary needed new shoes .	mary got new shoes .
	she followed the instructions as best as she could .	she executed the instructions as best as she could .
	pam wanted to have a special cake for her son 's birthday .	pam decides to have a special cake for her son 's birthday .
Sentiment	the food 's ok , the service is among the worst i have encountered .	the food 's wonderful , the service is among the finest i have encountered .
	we will not be using this location again .	we will definitely be seeking this location again .
	good selection of parts and accessories and reasonable prices .	poor selection of parts and accessories and high prices .
	it is a cool place , with lots to see and try .	it is a stupid place , with nothing to see and try .

Table 7: Samples of prompted sentiment controlled generations, using our Mix and Match LM and PPLM.

	Ours (Mix and Match LM)	PPLM
Pos Sent.	the country is noted for attracting a quarter-million tourists.	the country's top cycling event is right behind the olympics, and the
	the lake we come across can be said to be beautiful.	the lake is a great spot for swimming, diving and snorke
	the chicken and all the other ingredients produced a delicious meal.	the chicken wing is one of the best foods you can eat and it
	the movie was family-friendly and a success in japan.	the movie, which is currently only the third the the the the the
Neg Sent.	the country was unstable and was not ready to modernize.	the country's top animal welfare agency, the ministry of agriculture and food
	the lake was not supposed to be navigable under any circumstances.	the lake, a large, and the most massive and most terrible of
	the chicken was growling and beginning to feel a little sick.	the chicken noodles are the most horrible food i have ever had.
	the movie received only two nominations and earned no grand prix.	the movie is not in the , a, a, a

though we could have used any pre-trained attribute classifier from a model repository like Huggingface for E_{disc}, we train our own classifier for controlled empirical comparison. As described later, we do use pretrained Huggingface attribute classifiers as external attribute classifiers for fair evaluation against baselines. For experiments in which we add the BertScore (Zhang et al., 2020) component to the energy, we download the pre-trained `roberta-large_L17` models from Huggingface, respectively. We have provided implementation details and hyperparameter ablations of all the experiments in Appendix B.6, B.7, B.8 and B.9.

B.3 Baselines

PowerTransformer. For the task of controllable debiasing (agency revision), we compare our work with PowerTransformer (Ma et al., 2020), an approach that uses paraphrasing and self-supervision based on a reconstruction loss, building on pre-trained language models, to re-write text and control agency level of sentences.

He et al. For style transfer on sentiment an formality domains, we compare our work with He et al. (2020), a generative style transfer framework which uses a variational autoencoder (VAE) built using a sequence-to-sequence LSTM-based model to do unsupervised style transfer. This framework needs to be trained from scratch for each style transfer task.

UNMT. As a second baseline for style transfer, we compare our work with UNMT (Lample et al., 2018), an unsupervised machine translation

framework that demonstrates high performance for sentiment transfer.

PPLM. For the task of sentiment controlled generation, we compare our work to Plug-and-Play LM (PPLM) Dathathri et al. (2020), which does attribute controlled generation using the flow of gradients from discriminators trained on the last hidden layer representations of the generator, to guide generation.

FUDGE. This approach (Yang and Klein, 2021) trains step-wise discriminators on partial generations from GPT-2 to determine whether the constraints related to desired attributes will be satisfied by the future completion of the sequence or not. We compare against this on topic controlled generation as this approach was shown to be superior to PPLM on this task.

B.4 Evaluation Metrics

We use a variety of evaluation metrics to compare our approach's performance on two major facets: (1) Quality of generated text, and (2) success on matching the target attribute used for control.

B.4.1 Text Quality and Semantic Similarity

GPT-2 PPL. We feed our generated test sentences to a Huggingface (Radford et al., 2019) pre-trained GPT-2 xl model, and report its perplexity (PPL), as an automatic measure of fluency. Although this measure is not a perfect indicator of fluency, we find it to be a useful metric alongside human judgements. [3]

---

[3]Due to the high variance in the PPL scores generated across sentences by GPT-2, we report the median score for

12

Table 8: Samples of prompted topic controlled generations, using our Mix and Match LM and FUDGE.

	Ours (Mix and Match LM)	FUDGE
Computer	to review, please link to (chessworld.net/chessworld/download.html). in summary, key program clients are homeforge, blogdev and skeptic.net. it has been shown using several techniques, including microscopy, electron microscopy, and digital loansharking.	to review, instead of using the "n/a" flag (like on our previous posts) in summary:- install and run a local mysql server on the host computer- add a mysql table it has been shown using ebpf/ebpis (extraction of a new ebp
Legal	the connection to the assault was not without controversy, especially given the expert testimony the prosecutor had provided. to review, or submit information to the cdu regarding the current (constitutionally) electoral law. to conclude, when a claim is not true, the defendant's claims are often not true.	the connection failed, however, under an audit of one of the two, the judge said. the to review, the court's decision not to review the case raises an important question. the court's to conclude, the court held a motion is properly made to dismiss a claim for an award of attorney
Military	foundational to this is the cold war, which eliminates all military defense available to the enemy. views on the civil war fleet, the national maritime museum. views on the royal navy, admiralty. to conclude, we all agree that constructive defense methods are not yet available.	foundational to this is an attack on the conventional wisdom on the left that the left is the party views on russia's military buildup on the strength of his repeated insistence, a number of constructive defense? to conclude, the russian navy's attack on the malaysian ship, a taskforce carrying out exercises,
Politics	an illustration of: the historical background, culture, and general political significance of the books' contents. the issue focused on socialism, democracy, social justice and self-government in countries across the globe. in this essay, king frederick iii of prussia was prominently featured in american post-civil war culture.	an illustration of an anti-democratic regime under a fascist dictatorship and its suppression of the popular opposition and the issue focused on religious freedom in the country's constitution, a fundamental pillar of u.s. in this essay, the term "political correctness" is used to refer to political demands imposed on the
Religion	the issue focused on the inferiority of conservatives ("" religious conservatives "") vs . atheists . to summarise accurately the wind direction, additional characters may be added to the classification table below. an illustration of the natural history of wales by francis bacon. bateson, charles (1839).	the issue focused on religious freedom, particularly when the bible teaches that god is "the creator." to summarise, if the present-day christian churches were a monastic order of the monks rather an illustration of an ancient bronze age village in the northern greek region of crete, which shows a
Science	prior to this date, the manuscript was not currently available on the internet, and is cited rarely. the relationship has inspired research into the role of women in economics, and contributions to feminist economic theory. the issue focused on developments in the field of "darwinism, biology and human evolution" research.	prior to this experiment, the scientists had not seen a new species in the area since the late 1800 the relationship between energy use and energy use as a function of time was also investigated using a linear mixed the issue focused on data retention, and the key elements of the retention matrix, including retention of identifiers
Space	furthermore, the performance space is "packed with classical music" and is "lavishly decorated". to conclude, an asteroid becomes, mathematically, the largest asteroid to ever be "discovered". to summarise other countries'respective territorial claims, including territorial waters, islands, etc. .	furthermore, the eighty-first star is the planet's largest moon and it sits directly in between to conclude, scientists behind spacemonkey, and a number of the other projects that nasa is supporting to summarise: x (1x a2 a19 a1 a2 b2

Table 9: Sentiment transfer on Yelp dataset ablation study. The tuples in the first column show the (α, δ, β) set of parameters. We ablate the effect that different components have on the transfer. The *(ref)/(src)* next to the metrics denotes that they are measured with respect to the reference/source text. *Int./Ext. Clsf.* show the accuracy of the discriminator used in the energy/external discriminator from Huggingface. *Hamm.* shows the Hamming distance.

(Disc, MLM, Hamm.)	BLEU	GPT-2	BertScore	Hamm.	Int. Clsf.	Ext. Clsf.
(1,0,1)	4.77	1611.8	0.88	5.308	81.7	67.4
(1,0,0)	1.12	3825.3	0.85	8.378	99.0	84.5
(0,1,0)	3.77	101.3	0.90	5.92	24.7	29.3
(100,1,0)	2.89	143.0	0.88	7.067	99.2	96.5
(0,1,50)	23.60	110.0	0.99	0.002	4.3	5.0
(100,1,50)	19.71	191.5	0.95	1.838	94.7	82.8

BLEU. For sentiment (Yelp) and formality (GYAFC) transfer experiments, since we have reference text, we report the BLEU score. For controlled debiasing, we report BLEU between generated text and source, and show it as BLEU (src).

BertScore. As a measure of meaning preservation, we use the F1 BertScore metric (Zhang et al., 2020)

each system under comparison.

Table 10: Formality transfer on GYAFC dataset ablation study. The tuples in the first column show the (γ,η) set of parameters. We ablate the effect the BLEURT and BertScore experts have on the transfer. The *(ref)/(src)* next to the metrics denotes that they are measured with respect to the reference/source text. *Int. Clsf.* shows the accuracy of the discriminator used in the energy, and →*Informal/Form.* shows the breakdown of the external classifier accuracy. *Hamm.* shows the Hamming distance.

(BLEURT,BertScore)	BLEU	GPT-2	BertScore	Hamm.	Int. Clsf.	→Inf.	→Form.
(100,0)	14.07	243.9	0.87	5.93	89.34	97.41	19.80
(300,0)	13.75	233.9	0.88	5.88	89.34	97.01	22.94
(0,100)	17.78	206.3	0.89	5.22	91.15	96.67	23.13
(0,300)	18.85	210.9	0.90	4.91	88.23	97.04	23.13

to compare the semantic similarity of the provided reference sentence with the generated output.

Hamming Distance. We also report the hamming distance between the source text and generated text, to measure the extent of the change induced by our framework.

B.4.2 Attribute Quality

Internal Classifier Accuracy. To evaluate the quality of applying target attributes, we report accuracy of the internal classifier (the discriminator used for generation) on the generated text, assuming the target attribute is the correct label. The higher this accuracy is, the better.

External Classifier Accuracy. Since the internal classifier is the one we are sampling from, it is natural that we would get high accuracy on it, compared to our baselines. To create a more fair comparison, we also report classification accuracy using external classifiers, downloaded from Huggingface. For sentiment classification we use `textattack/bert-base-uncased-yelp-polarity` (Morris et al., 2020), and for formality we use `cointegrated/roberta-base-formality`.

Agency Lexicon Accuracy. For the controlled debiasing experiment, we measure the accuracy of the change in agency by comparing the target agency level with that of the generated text, extracted using the connotation frames lexicon, and following the setup from Ma et al. (2020).

B.5 Hyper-parameter and Component Selection

Selection of components is based on the needs of the task and is straight forward. You add each component you need, to satisfy some condition. If you want to do sentiment controlled generation, you add a sentiment classifier. Finding the hyper-parameters for each component (the multiplier in energy) is also simple, since the trade-off between

the different components is clear. For instance, as shown in Table 9, increasing the discriminator score results in a more successful sentiment transfer, and increasing the Hamming score results in keeping the sentence the same.

B.6 Controllable Debiasing: Hyper parameters

For the results presented in Table 1, we ran the Gibbs chain for 8 epochs (8 iterations over all the tokens) for the conventional mode of our method, and 30 iterations for verb replacement. We used the parameters $\alpha=100,\beta=50,\theta=100$, where θ is the coefficient assigned to the agency scorer, and α and β are defined in Equations 1 and 2.

B.7 Sentiment Transfer: Hyperparameters

In this section we discuss the hyperparameters used for sampling and see the effects of each one. For the results presented in Table 2, we ran the Gibbs chain for 8 epochs (8 iterations over all the tokens), and used the parameters $\alpha=100,\beta=25$ (for Discriminator ↑) and $\alpha=100,\beta=50$, for Hamming ↑. α and β are defined in Equations 1 and 2.

Table 9 shows six different scenarios, with six different coefficeints for the Disciriminator (α), BERT MLM (δ) and Hamming distance (β) components in the energy function, which helps understand the effect each expert has.

B.8 Formality Transfer: Hyperparameters

For the results presented in Table 3, we ran the Gibbs chain for 5 epochs (5 iterations over all the tokens), and used the parameters $\alpha=140,\beta=15,\gamma=100$ (for Discriminator ↑) and $\alpha=140,\beta=50,\gamma=300$, for BertScore ↑. α, β and γ are defined in Equations 1 and 2.

Table 10 shows four different scenarios, with four different coefficeints for the BLEURT and BertScore components in the energy function, which helps understand the effect each expert has. For BLEURT, we use pre-trained

`Elron/bleurt-base-512` from Hugging-face.

B.9 Prompts and Hyperparameters Used for Controlled Generation

We have listed the prompts that we used for controlled text generation (these prompts are taken from Dathathri et al. (2020)): the country, the lake, the chicken, the movie, the pizza, the painting, the year, the city, the book, the potato, the horse, the road, the president, once upon a time. We collect these prompts from PPLMs github repo, available at this url: `https://github.com/uber-research/PPLM/tree/master/human_annotation/pplm_labeled_csvs`.

PPLM has multiple knobs to tune for sampling, and after running a greed search we found that `gamma=1,num_iterations=10,step_size=0.1,kl_scale=0.01` and `gm_scale=0.95` yeild the best results (reported in Table 4). We generated samples by running the command `python run_pplm.py -D sentiment`, with the mentioned hyperparameters. For FUDGE, we tune the λ parameter, and we find that $\lambda = 10$ works best.

For our method, we ran the Gibbs chain for 15 epochs, and used hyperparameter $\alpha = 40$, from Eq. 1. We don't use any experts other than the yelp sentiment classifier, so we don't have any other hyperparamters.

B.10 Human Evaluations

We used Amazon Mechanical Turk for our evaluations, where each HIT was a two choice question of "which sentence is more fluent?" and the providers were paid $0.1 per HIT. We selected Turkers from English speaking countries. We also had each each question answered 3 times (by 3 Turkers), to create redundancy and robustness.

B.11 GPU Hours and Infrastructure

One of the main purposes of this work is to introduce a paradigm in which we re-use existing models and do not retrain. As such, we did not need GPUs for training (we finetuned two classifier for demonstration purposes, which took less than two GPU hours).

However, we do use GPUs for inference (less computationally intensive), for generating samples. We used an in-house 4GPU server (NVIDIA RTX2080), and the samplings and hyperparameter tuning took an overall of around 10-14 full days on the 4 GPUs.

15