We thank all reviewers for their detailed constructive feedback and suggestions. Below are our responses:

Clarify Technical Contributions (R3 / R4):

- **Gradient Estimation**: While the reviewers point out that REINFORCE and Gumbel-Softmax are now well-established techniques for discrete Categorical distributions (fixed N sample space), when extending them to *structured / CRF* distributions (combinatorial complexity, N^T sample space, T=sentence length), there are still many open challenges (e.g: seq-level. v.s. stepwise grad. Section 4, Appendix B.2; recurrent v.s. immediate grad. as pointed out by reviewer 2). We note that there is an active interdisciplinary effort on this task from multiple communities (NLP [1], Optimization [2], Probabilistic ML [3]). Our paper makes a targeted contribution about gradient structures in linear-chain CRFs for text generation and shows a novel use of Gumbel-Softmax for structured models.
- **Practical Benefits**: Training structured variables with REINFORCE is notoriously difficult [4], and Gumbel-CRF substantially reduces the complexity. Table B (below) demonstrates this empirically. Gumbel-CRF has: (a) fewer hyperparameters to tune; (b) less sensitivity to random seeds; (c) better gradient estimates; (d) less posterior collapse, especially for structured inference models (large amount of efforts in our experiments were to use multiple tricks to make REINFORCE work without posterior collapse). Table A (discussed below) further shows that these benefits persist in an auto-regressive setting. These advantages would considerably benefit all practitioners (just as Gumbel-Softmax has) with significantly less training time and resource consumption.

Additional important concerns:

- **NLL with importance sampling / autoregressive decoder (R2 / R4)**: We agree that using word dropout / non-autoregressive (NAR) decoder is not clearly motivated and orthogonal to the contribution of the paper. (This was originally done to compare with other latent-variable learning approaches.) Table A (below) gives the results for an autoregressive (AR) decoder for text modeling with NLL estimated by importance sampling. (ELBO results in the main paper Figure 3 did not add the constant term C in Equation 10, so we repeat the comparable NAR results here.) These experiments show that when trained with Gumbel-CRF, the AR decoder outperforms REINFORCE.
- **Clarification of terms and algorithms (R2)**: We apologize that using the term "differentiable z" may give the wrong implication as Argmax is differentiable almost everywhere [3]. We will clarify this in the paper. In terms of a full relaxation with the recurrent part, it could be implemented by changing line 7 in Algorithm 2 to an expectation weighted by $\tilde{z}_{t+1}$. Because we use a straight-through estimator, we want to recover an exact sample $\hat{z}$ with the Argmax in line 9 for the forward pass. In structured models, a fully relaxed sample path may diverge from the exact sample path. Our current relaxation couples the two. We will add more detailed discussions.
- **Modeling details (R1 / R3 / R4)**: For the text modeling experiments, we use the same underlying model for PM-MRF and Gumbel-CRF. For the straight-through estimator, we use the hard sample $\hat{z}$ in the forward pass, and the soft sample $\tilde{z}$ in the backward pass (source code torch_model_utils.py, line 293). For data-to-text generation, we use semi-Markov models as our baselines as had been done in previous work (Table 2). During testing, given a key-value pair, we find the training instance with the closest key under Jaccard distance, and use their templates for the given test case. For paraphrase generation, given a sentence, we retain its BOW, and retrieve a template from the training set with the closest BOW under Jaccard distance, and generate a new sentence. To get the segmentation in Figure 4, we collapse consecutive states into one state index, and report the state ngrams.
- **Comparison to SOTA models on paraphrasing and data-to-text (R3)**: Our method is orthogonal to many of the additional modeling techniques in SOTA models (e.g. posterior regularization in the SM-CRF model) so they can be integrated with ours. Although our model does not outperform existing SOTA models that use specifically designed techniques for each task, we aim to show that the approach scales and has auxiliary benefits (discussed above).
- **Comparing REINFORCE and Gumbel-CRF (R4)**: We unfortunately are not able to run new experiments comparing BLEU/ROUGE with REINFORCE. We believe with careful tuning, a REINFORCE model may perform well. However, even if the two perform similarly in metrics, the Gumbel-CRF shows significantly reduced modeling complexity (fewer hyperparameters and less sensitive to random seeds), making it a beneficial approach.

	Dev		Test	
	Neg. ELBO↓	NLL↓	Neg. ELBO↓	NLL↓
REINFORCE NAR	73.85	73.82	73.81	73.79
Gumbel-CRF NAR	69.34	66.42	69.49	66.94
RNNLM	–	35.15	–	34.45
REINFORCE AR Dec	36.53	35.93	35.74	35.18
Gumbel-CRF AR Dec	36.85	34.95	35.87	34.19

Table A. NLL Comparison

	Optimized Hyperarameters	Dev NLL (6 random seeds under same hyper params.) mean	max	min	std
Gumbel-CRF NAR	Entropy regularization Softmax temperature	68.28	66.42	69.23	1.01
REINFORCE NAR	Entropy regularization Constant baseline Baseline model Reward scale Number of samples	79.54	73.82	94.37	7.05

Table B. Hyperparameter Comparison and Sensitivity Analysis

[1] Hao et al. Backpropagating through Structured Argmax using a SPIGOT. ACL 2018

[2] Mensch and Blondel. Differentiable Dynamic Programming for Structured Prediction and Attention. ICML 2018

[3] Paulus et al. Gradient Estimation with Stochastic Softmax Tricks. Arxiv 2020

[4] Kim et al. Unsupervised Recurrent Neural Network Grammars. NAACL 2019