We thank all reviewers for the valuable comments. First, we respond to two common concerns raised by the reviewers, and then answer other questions raised by each reviewer.

[Comment: By augmenting the decoder with a CRF, the resulting network is no longer non-autoregressive.]

We will use a new title "Structured Decoding for Fast Machine Translation", as suggested by Reviewer #5. We will also improve our presentation in other related parts of our paper. However, we still want to emphasize our speedup comparing to vanilla autoregressive models.

[Comment: In Equation 2 and 3, the variable z is unbound, and its explanation as "a well-designed input z" is insufficient for its understanding.]

This is a general formula for non-autoregressive machine translation. In our NART, z is deterministic, but in other models such as (Gu et al., 2017) or (Roy et al. 2018), z is stochastic. For example, in (Gu et al.), z is a fertility-related copy of source tokens, while in (Roy et. al), z is a sequence of autoregressively generated discrete latent variables. We will make our presentation more clear in the final version. Specifically, we will fix the typo in Equation 3: n should be T' over the Σ mark.

To Reviewer #3 Thanks for the detailed review! Regarding your comments: **(1)** *"A fast beam-search using a tri-gram language model may already give similar improvements."* This is a possible approach to extending our current practice. However, we would like to emphasize that although a linear-chain CRF is similar to a bi-gram language model, it has many other advantages such as **end-to-end** training with neural networks, as well as **exact inference** over the labeling/translation, as pointed out by Reviewer #4. Moreover, the autoregressive part should be simple enough to keep small overhead. **(2)** We describe the hardware setting at L224. **(3)** Since the target length is fixed, we try different target lengths ranging from $(T + C) - B$ to $(T + C) + B$ (at L216, Sec 4.2). Therefore, we set B as 4 and 9 for rescoring 9 and 19 candidates. Rescoring 9 is for a fair comparison with previous work, while rescoring 19 is to explore the limits of our model. We will improve our presentation about this in the final version and include descriptions of how other models rescore their candidates. **(4 - L40)** Note that the examples in (Gu et al., 2017) also show this type of translation errors. This inconsistent pattern is common in NART literature and we will add references about this type of translation error. The examples shown in Table 1 are from our implemented models (as described in Sec 4 of the paper). **(5 - L214)** In order to show the effectiveness of the structured inference module, we choose the simplest target length prediction model. Please note that the precision of the predicted target length becomes less important when we rescore more candidates. Thank you for your suggestion. Actually, we have already noticed this problem in the future work section (L272) and plan to improve the target length prediction module in our future work. **(6 - L222)** We follow common practice in previous works to make a fair comparison. Specifically, we use tokenized case-sensitive BLEU for WMT datasets and case-insensitive BLEU for IWSLT datasets. We will include this in the final paper. **(7)** Regarding other comments, please refer to the general response sections for L38, and we will fix the issues at L34, L59 and L140 in the final version. We appreciate your detailed review comments and hope the response address your concerns.

To Reviewer #4 Thanks for your positive comments! Regarding your suggestions: **(1)** *"I think the main promise of this approach is in exact decoding, though the authors do not investigate this much."* Thank you for your comment. We will provide more discussion about this direction in the final version. **(2)** $O(nk^2)$ complexity is the FLOPs (mul-adds) of the unparallelizable part of CRF module. In comparison, the time complexity of an one-layer RNN-based decoder without attention mechanism is $O(nh^2)$, where n is sequence length, h is the number of hidden units and h is usually several times larger than k. We will provide more discussion about the time complexity in the final version. **(3)** According to your suggestion, we conducted a set of experiments and here are the results (LSTM-based is from a 8-layer modification of `lstm_attention_base` in `tensor2tensor` library, which has similar model size with `transformer_base`):

Models	WMT14 En-De	Latency	Speedup
LSTM-based (beam size = 1)	/	2031.6*ms*	/
Transformer (distilled, beam size = 1)	26.48	240*ms*	1.61×

To Reviewer #5 Thanks for your positive comments! Please refer the previous general response section about some of your concerns. **(1)** For the use of different target lengths, we tried different C and find that the current setting yields the best performance. **(2)** The results of ablation study on the effect the vanilla non-autoregressive loss:

λ	0.0	0.1	0.5	1.0
WMT14 En-De (NART-CRF, no rescoring)	fail	22.42	**23.32**	22.59