We thank all reviewers for the valuable feedback. We will fix the formatting issues and address the concerns pointed out by the reviewers in the final manuscript.

Response to Reviewer 1:

High rank with $K = 1$: In our preliminary experiments, using $K = 1$ resulted in degraded performance. We believe this is because (1) $K = 1$ reduces the empirical rank and (2) using a large K imposes a "branching" inductive bias that benefits training. We will compare the performance and empirical rank of different K's in our revision.

High-rank argument: We believe this is a very good point. We will tone down about the high-rank argument that elementwise multiplication leads to high-rank representations because the argument is mainly empirical, though somewhat intuitive. We will also add a study about the empirical rank in our revision.

Response to Reviewer 2:

Line 117: The argument is mainly empirical. Intuitively, it is likely that the results of elementwise multiplication are high-rank because the features are randomly distributed. We will clarify this and add a study about the empirical rank in our revision.

Line 157: "partially high-rank" means that frequent tokens have high-rank representations and infrequent tokens have low-rank representations (lines 159-161). We will further clarify this.

Line 121: It is an open question why unnormalized priors do not work. We conjecture that there might be two reasons: (1) normalization introduces competition among different branches, which encourages different branches to learn different features; (2) normalized inputs have more stable norms, which might lead to more stable optimization. This is similar to the attention mechanism where normalizing the attention features with probabilities summing to one is necessary for the best performance. We believe these are very important points. However, as understanding and improving prior normalization (e.g. using L1 normalization) is largely nontrivial, we leave them to future work.

Hyperparameters: For the value r, we try $r = 0.1$ and $r = 0.5$. The performance of $r = 0.5$ is slightly better than or equal to $r = 0.1$, but we found the gains of using $r = 0.5$ are small for LM in preliminary experiments so we use $r = 0.1$ for LM. We fix the Gaussian noise at 0.1 for all experiments. The other hyperparameters are shared by MoS and Mixtape and we use the same hyperparameter search space for the two methods. We performed random search and ensure the same number of trials are used for the two methods. The search space includes: dropout [0.0, 0.1, 0.3], learning rate [0.1, 0.2]. We will include the numbers and clarify the settings in our revised paper.

Comparison with [9]: We mainly focus on improving the efficiency in this paper. Because [9] is more expensive than MoS, we use MoS as our direct baseline. We will include [9] in our tables to give readers more information.

Other possibilities: We believe that the ideas of hierarchical softmax and word clustering are appealing, which are interesting directions for future work. We will also include the related work in our updated version.

Code: We will publish our code for reproducing all of our results in this paper.

Response to Reviewer 3:

Background: We will add more details to the background section, better explaining 'softmax bottleneck' and 'mixture of softmaxes'.

Hyperparameters and ablation study: We believe it is valuable to perform a study to understand the effects of different values of K and r. In our experiments, we fix $K = 4$ throughout all experiments to minimize the effects of hyperparameter tuning, and this value is recommended for future use of our method. We experimented with $r = 0.1$ and $r = 0.5$ in our early experiments, and found the two values have similar performance on language modeling, while using $r = 0.5$ yields improvement of about 0.1 to 0.3 BLEU for machine translation. As a result, we believe our model is not very sensitive to these hyperparameters and the default values will be sufficient for most tasks. We will provide more detailed analysis and comparisons in our final version of the paper.

Why Mixtape: We use this name because the "mix" prefix is related to our approach of mixing the logits.