Effective Exploitation of Posterior Information for Attention-Based Speech Recognition

JIAN TANG\textsuperscript{1}, JUNFENG HOU\textsuperscript{1}, YAN SONG\textsuperscript{1}, LI-RONG DAI\textsuperscript{1}, (Member, IEEE), AND IAN MCLoughlin\textsuperscript{1,2}, (Senior Member, IEEE)

\textsuperscript{1}National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230026, China
\textsuperscript{2}School of Computing, University of Kent, Medway, Canterbury CT2 7NZ, U.K.

Corresponding author: Jian Tang (enjtang@mail.ustc.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1002202, in part by the National Natural Science Foundation of China under Grant U1836219, and in part by the Key Science and Technology Project of Anhui Province under Grant 18030901016.

\textbf{ABSTRACT} End-to-end attention-based modeling is increasingly popular for tackling sequence-to-sequence mapping tasks. Traditional attention mechanisms utilize prior input information to derive attention, which then conditions the output. However, we believe that knowledge of posterior output information may convey some advantage when modeling attention. A recent technique proposed for machine translation called the posterior attention model (PAM) demonstrates that posterior output information can be used in that way for machine translation. This paper explores the use of posterior information for attention modeling in an automatic speech recognition (ASR) task. We demonstrate that direct application of PAM to ASR is unsatisfactory, due to two deficiencies; Firstly, PAM adopts attention based weighted single-frame output prediction by assuming a single focused attention variable, whereas wider contextual information from acoustic frames is important for output prediction in ASR. Secondly, in addition to the well-known exposure bias problem, PAM introduces additional mismatches in attention training and inference calculations. We present extensive experiments combining a number of alternative approaches to solving these problems, leading to a high performance technique which we call extended PAM (EPAM). To counter the first deficiency, EPAM modifies the encoder to introduce additional context information for output prediction. The second deficiency is overcome in EPAM through a two part solution of a mismatch penalty term and an alternate learning strategy. The former applies a divergence-based loss to correct the mismatch bias distribution, while the latter employs a novel update strategy which relies on introducing iterative inference steps alongside each training step. In experiments with both WSJ-80hrs and Switchboard-300hrs datasets we found significant performance gains. For example, the full EPAM system model achieved a word error rate (WER) of 10.6% on the WSJ eval92 test set, compared to 11.6% for traditional prior-attention modeling. Meanwhile, on the Switchboard eval2000 test set, we achieved 16.3% WER compared to the traditional method WER of 17.3%.

\textbf{INDEX TERMS} Speech recognition, posterior attention, divergence penalty, exposure bias, alternate learning strategy.

\section{I. INTRODUCTION}
Automatic Speech Recognition (ASR) has achieved tremendous progress over recent years, gradually evolving from the original hybrid architecture \cite{1}–\cite{3} to end-to-end systems and models \cite{4}–\cite{7}.

Among the latter, attention-based sequence-to-sequence models have achieved significant improvement over conventional ASR systems \cite{8}, \cite{9} and are thus a popular research direction. In attention-based sequence-to-sequence (seq2seq) models, the chain rule is usually used to decompose the sequence mapping problem into a multiplication of recursive elements. The advantage of attention models is that they can directly learn a mapping from speech to text, which enables joint optimization of acoustic and language models \cite{10}, \cite{11}. There are many variants of attention architectures, including soft \cite{12}–\cite{15}, hard \cite{16} and variational \cite{17} but all work, in an encoder-decoder architecture,
by focusing the attention of the decoder on specific parts of the encoder output which are more important to system output.

However, there are still some limitations that affect the performance of such systems. One is that the above systems use prior input information to compute an attention vector, and do not take account of system output. We already know that posterior information from the previous output step plays a key role in model classification, and it can also benefit sequencing, like in a language model. Besides this, posterior information can be further applied to the sequence alignment process. Since attention reflects an alignment of the input and output, knowledge of posterior information is likely to be beneficial in improving the accuracy of the alignment distribution.

Recently, Shankar and Sarawagi proposed a posterior attention model (PAM) [18] for neural machine translation which demonstrated that utilization of posterior information in this way is beneficial for sequence-to-sequence modeling. They began with an explicit joint distribution of all posterior outputs and attention variables in a prediction sequence. PAM was found to be effective at utilizing posterior information, achieving better BLEU scores [19] and more accurate alignment for their application area of machine translation. In this paper, we aim to use posterior information for attention modeling in ASR and thus begin by applying PAM. However, we find that its performance in ASR is overly affected by the authenticity of posterior information, effectively negating its benefits in practice.

PAM makes use of posterior information not only for classification, but also for sequence alignment. While it works well for machine translation despite these potential side effects, do the longer input sequences and the difficulty of distinguishing similar speech fragments affect alignment optimization in ASR? Moreover, does the hypothesis that posterior information is useful still hold for ASR, or is it even more relevant? To answer these questions, we explore methods of obtaining the advantages of posterior information for forming attention in ASR, but without the disadvantages.

A discrepancy between ground-truth labels used for training, and the historical posteriors used during inference, means that predicted words at training and inference are drawn from different distributions (specifically the data distribution and model distribution respectively). This is called exposure bias [20]. During training, the decoder is conditioned on ground-truth prefix tokens, while hypothesized ones are used during inference. Moreover, apart from the effect of historical tokens in the standard architecture [21], PAM additionally uses the current output to condition the posterior alignment distribution, introducing a further mismatch between training and inference. Together, this discrepancy and mismatch cause significant degradation in performance during inference.

This paper addresses each of those limitations noted above, and presents a unified end-to-end ASR system that effectively exploits posterior information, without the negative effects; we call this the Extended Posterior Attention Model (EPAM).

Specifically, we use an original posterior attention framework to obtain accurate alignment, then combine novel methods to correct for exposure bias. For alignment optimization, we directly limit its distribution, rather than prescribe a limit to the posterior information. A divergence-based penalty loss is introduced to ensure the obtained posterior alignment distribution is similar to the target distribution when prediction estimation errors inevitably occur. As for the use of posterior information during classification, we propose a method we call the alternate learning strategy (ALS) to mitigate the exposure bias problem that commonly exists in sequence-to-sequence models [22]–[25]. ALS requires that, at each decode step, the current output prediction is found in the teacher forcing setup, and then a subsequent $M$ auxiliary inference steps recursively refine that prediction. Losses from the two types of prediction are applied alternately to update the model during training.

In summary, the system presented in this paper has the following three novel contributions. Firstly, we first evaluate the posterior attention model for a speech recognition task and, finding it problematic, propose several modification to its structure. The resulting system performs well, achieving 11.5% word error rate (WER) on WSJ [26], against baseline scores of 12.0% WER. Secondly, we propose and evaluate a divergence-based penalty to control the use of posterior information for model optimization. This further improves WER to 11.1% (without language model rescoring), outperforming both a soft-attention baseline (12.0% WER) and the original PAM by a significant margin. Next, the novel ALS update strategy is proposed to reduce the dependence of ground-truth posterior information during training. Incorporating ALS improves WER performance from 11.5% to 10.9%. Combining the three proposed modifications into the final EPAM system, we achieve a WER performance of 10.6% against our baseline of 12.0%. EPAM easily outperforms the traditional soft-attention system with schedule sampling (SS) [25], which achieves 11.6%.

The remainder of this paper discusses related work in Section II, describes the proposed modifications and their motivations in detail in Section III, evaluates them in Section IV and concludes the paper in Section V.

II. RELATED WORK

A. ATTENTION-BASED SEQUENCE-TO-SEQUENCE

A seq2seq model is an end-to-end neural network that maps a dynamic length sequence $x = (x_1, x_2, \ldots, x_L)$ of length $L$ to another dynamic length sequence $y = (y_1, y_2, \ldots, y_T)$ of length $T$. Given an input sequence $x$, the goal is to model the conditional distribution $P_{\theta}(y|x)$ of output sequence $y$. Each $y_t$ depends not only on other tokens in the sequence, but in practice, it usually depends mainly on some specific focused part of the input sequence. A hidden variable $\alpha_t$, called the attention variable, denotes which part of $x$ the output $y_t$ depends on. The set of all attention variables is $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$. During training the input $x$ and output $y$ are observed but the attention $\alpha$ is hidden. So the conditional
distribution can be further expressed as (1),

\[
P_0(y|x) = \sum_{\alpha} P_0(y, \alpha|x) = \prod_{t=1}^{T} f P_0(y_t|c_t, s_t, y_{t-1})
\]

(2)

Here, we show the commonly-used form of a seq2seq model as proposed originally by Bahdanau et al. [27]. There are three main parts to this model: encoder, attention, and decoder modules. The encoder processes input sequence \( x \) to produce a high level representation encoded in the continuous vector \( x_{1:L} \). The attention bridges the information between encoder representation \( x_{1:L} \) and the current decoder’s states \( s_t \). Given a pair of encoder and decoder states, the attention scoring module finds an alignment between each element of the output sequence and the hidden states generated by the acoustic encoder network for each acoustic input frame, and produces a weighted-sum of state sequences which serve as relevant context \( c_t \) for the decoder. By learning an alignment between acoustic representations and output labels, the attention mechanism can select the most related content from the input sequence. Clearly, the accuracy of the alignment directly affects the classification performance of the model.

The final seq2seq component is the decoder module which generates the target discrete sequence \( y \). The conditional probability \( P_0(y|x) \) is further decomposed via the chain rule as shown in eqn. (2). It can be seen from the formula that, at each output position, the decoder generates an output discrete sequence \( y_t \) based on both historical token \( y_{t-1} \), the attended representation \( c_t \) and the decoder state \( s_t \). In this module, the employment of historical prediction fundamentally leads to a discrepancy between the training and inference stages, since training uses ground-truth labels to update the system, whereas during inference, only estimated labels are available.

B. POSTERIOR ATTENTION MODEL (PAM)

To obtain more accurate and reasonable seq2seq alignment, a method called the posterior attention model (PAM) was recently proposed for machine translation [18]. PAM factorizes eqn. (1) via the chain rule, but jointly with hidden alignment variable \( \alpha \), and \( y \). With full consideration of all possible latent-alignment distributions, the conditional probability \( P(y|x) \) can then be rewritten as follows (here \( x \) is dropped to use the shorter form \( P_0(y) \) for \( P_0(y|x) \)),

\[
P_0(y) = \prod_{t=1}^{T} \sum_{\alpha_t} \sum_{y_{t-1}} p(y_t|y_{t-1}, \alpha_t) p(\alpha_t|y_{t-1})
\]

Thus, the conditional probability has been expressed as a product of factors that apply at each time step \( t \), while conditioning only on previous outputs and attention. The term \( \sum_{\alpha_{t-1}} p(\alpha_t|\alpha_{t-1}, y_{t-1})p(\alpha_{t-1}|y_{t-1}) = P(\alpha_t|y_{t-1}) \) is the attention at step \( t \), conditioned on all previous outputs.

Since this is the attention distribution before observing the output label at the corresponding step, it is referred to as prior alignment, denoted as \( \alpha_t^{\text{prior}} \). Meanwhile, the probability of \( P(\alpha_{t-1}|y_{t-1}) \) is called the posterior alignment \( \alpha_t^{\text{post}} \) at step \( t-1 \). In eqn. (3), a joint distribution of output and attention should be calculated at each step and all the attention distribution cases need to be considered. During this process, the number of variables involved in this summation is large. So, to simplify the calculation, the attention variables \( \alpha_t \) are assumed to belong to one focused input (like in a hard attention mechanism). Under this assumption, the following calculation is used [18], where RNN represents one recursive neural network layer.

\[
\alpha_t^{\text{prior}}(a) = \sum_{\alpha_t'} P(a|s_{t-1}, a') \alpha_{t-1}^{\text{post}}(a')
\]

(4)

\[
P_0(y|x) = \prod_{t=1}^{T} \sum_{\alpha_t} \sum_{y_{t-1}} p(y_t|y_{t-1}, \alpha_t) p(\alpha_t|y_{t-1})
\]

(5)

\[
\alpha_t^{\text{post}}(a) = P(y_t|s_t, x_a) \alpha_t^{\text{prior}}(a) / \sum_{\alpha_t'} P(y_t|s_t, x_a) \alpha_{t-1}^{\text{prior}}(a')
\]

(6)

\[
s_{t+1} = \text{RNN}(s_t, y_t, \sum_{\alpha_t} \alpha_t^{\text{post}}(a)x_a)
\]

(7)

Compared to the basic seq2seq method described in Section II-A, two major differences can be observed. First, we note that the output of eqn. (5) is a mixture of multiple output distributions each of which is a function of one focused input (like in hard attention). \( \alpha_t^{\text{prior}} \) directly acts on posterior probability \( P(y_t|s_t, x_a) \) to obtain the final output prediction, while the attention in basic seq2seq modeling act on the input content fusion. The second difference is that the attention distribution that is propagated to the next step is posterior to observing the current output. A posterior-refined alignment is hypothesised to be more accurate than a prior-referenced one, since it has knowledge of the output token at the current step.

C. TEACHER FORCING AND EXPOSURE BIAS

Typically, seq2seq models are trained to maximize the conditional likelihood of the correct output symbols. To predict a current token, the previous token from a ground-truth sequence is fed as an auxiliary input to the decoder during training. This so-called teacher-forcing [28] method helps the decoder to learn an internal language model (LM) for the output sequences. However, it also introduces a major drawback. In practice, the model is trained given the current state of the model and the previous target label \( y_{t-1}^* \). However, during inference, true previous ground-truth targets are unavailable, and are thus replaced by its own previous predictions. In this way, the model is biased to only perform well on the ground-truth history distribution, which differs from the model-prediction distribution. Furthermore, as the target sequence grows, these errors accumulate along the sequence, and the model has to perform inference under conditions it has never met during training [29]. All of those lead to poor generalization which seriously affects seq2seq.
model performance. This discrepancy is usually referred to as exposure bias, since it introduces a bias toward target labels [20].

Several researchers have noted, and attempted to solve, the exposure bias problem in ASR and other seq2seq tasks [29]–[33]. One method is to increase the employment of the model predictions during training to ease the reliance on the target labels. For example, scheduled sampling handles exposure bias by sampling a context from the previous ground-truth and the previous predicted token with variable probability during the whole training process [25]. This method expects to find an equilibrium point between model convergence and the proportion of inference labels used during training. Although the method is easy to implement, it is not straightforward to obtain an appropriate $c_t$ value or determine the variable probability. In practice, it requires a trial-and-error approach.

In the section below, we propose a novel method to overcome the exposure bias problem. Like schedule sampling, this utilizes past predictions to refine the reliance on ground-truth labels during training, however it does not require lengthy trial-and-error tuning of operating variables.

III. EXTENDED POSTERIOR ATTENTION MODEL

A. GENERAL FRAMEWORK

Given a set of speech utterances, suitably parameterized into feature vector $x$, and the corresponding ground-truth label sequence $y^*$ (which could represent characters, phonemes or other tokens), the general framework is shown in Fig. 1. The encoder produces a high level representation encoded in the continuous vector $x_{1:L}$, and the decoder generates predicted output $\hat{y}$ by focusing on the relevant elements of the hidden state at each time step $t$. As in PAM, we divide the alignment distribution into $\alpha_{t \text{ poz}}$ and $\alpha_{t \text{ pri}}$ depending on whether the current prediction $y_t$ is used or not. To illustrate operation, consider the decoding process at the $t$-th step. EPAM first performs state update and alignment calculations. The decoder updates current state $s_t$ based on output and posterior alignment from the previous step, as shown in eqn. (8). The decoder state $s_t$ together with $y_{t-1}$ and $\alpha_{t-1 \text{ poz}}$ are further fed into the calculation of current alignment score $\alpha_{t \text{ pri}}$. Attend in eqn. (9) describe the most generic attention. Based on the characteristics of monotonous alignment in ASR, location-based attention is used here [8],

$$s_t = \text{RNN}(s_{t-1}, [y_{t-1}, \sum_a \alpha_{t-1 \text{ poz}}(a)x_a])$$ (8)

$$\alpha_{t \text{ pri}} = \text{Attend}(y_{t-1}, s_t, \alpha_{t-1 \text{ poz}})$$ (9)

where, $x_a$ represents the $a$-th element in the encoder representation, $\alpha_{t-1 \text{ poz}}(a)$ means the corresponding posterior alignment weight at previous time step.

The next step is to obtain an output prediction. In basic seq2seq, an attention score is calculated which directly acts on the encoder representation to obtain the context vector $c_t$ as the weighted sum of $x_{1:L}$. Then, the decoder updates its state to $s_t$ based on $s_{t-1}$ and $c_t$ and produces $y_t$. To conclude, while the basic seq2seq operational order is “weighting before prediction”, the corresponding order is reversed in PAM and EPAM. Each element in the encoder representation predicts an output at each step (single-frame prediction) in eqn. (10). Those posterior output probabilities are weighted by $\alpha_{t \text{ prior}}$ to obtain the final model predictive output (11),

$$p(y_t|x, s_t, y_{t-1}) = \text{softmax}(Wx_a + Us_t + Vy_{t-1})$$ (10)

$$p(y_t|x, s_t, y_{t-1}) = \sum_a \alpha_{t \text{ prior}}(a)p(y_t|x_a, s_t, y_{t-1})$$ (11)

where trainable parameters $W, V$ and $U$ are matrices, and the bias variable is omitted for clarity.

Finally, after obtaining the prediction $y_t$, the prior alignment distribution $\alpha_{t \text{ prior}}$ can be further refined to provide a more accurate alignment for the next decoding stage, in eqn. (12). The attention score, obtained by the weighted single-frame prediction output, contains the output knowledge $y_t$ so we denote this the posterior alignment vector $\alpha_{t \text{ poz}}$. The above formulae (8)-(12) constitute the complete.
decoding process at step $t$.

$$
\alpha^\text{post}_{t}(a) = f(\alpha^\text{prior}_{t}, y_t) = \frac{\alpha^\text{prior}_{t}(a) \cdot p(y_t|x_a, s_t, y_{t-1})}{\sum_{a'} \alpha^\text{prior}_{t}(a') \cdot p(y_t|x_{a'}, s_t, y_{t-1})} \quad (12)
$$

However, without global context for each element $x_t$, the single-frame prediction in eqn. (10) may not perform well. Besides, the decoder state $s_t$ contains previous alignment $\alpha^\text{post}_{t-1}$, rather than the current position. Even though strong correlation exists between $\alpha^\text{post}_{t-1}$ and $\alpha^\text{prior}_{t}$ in ASR, we still believe that the location of the current prior alignment is expected to be more accurate than the previous posterior one. Therefore, we propose the addition of an RNN layer between eqns. (9) and (10), to update the decoder state using information similar to the glimpse in [8],

$$
s_t = \text{RNN} \left[ s_t, \sum_a \alpha^\text{prior}_{t}(a)x_a \right] \quad (13)
$$

In our evaluation, we will assess model performance both with and without the optional RNN layer.

In addition to this, we propose specific adjustments to PAM to account for the unique characteristics of the ASR task. For example, compared to machine translation, ASR requires more input context information [3], [34], [35]. Derived from an assumption that each element in the encoder output sequence should produce its own token prediction (denoted as the single-frame assumption), which leads to a lack of context information in ASR token prediction.

In order to add contextual information for each element in hidden state $x_{1:L}$ to refer to the gated linear unit [36], we modify the encoder to append a context block, consisting of a convolutional layer and fully connected (FC) layer, followed by the original encoder. We denote this module the context block (abbreviated as Ctx), defined as,

$$
\tilde{x}_{1:L} = FC(x_{1:L}) \times \sigma(\text{Conv}_{k \times 1}(x_{1:L})) \quad (14)
$$

where $\sigma$ and $\text{Conv}_{k \times 1}$ represent sigmoid activation and a 1-D temporal convolution layer with kernel size $2k + 1$ respectively.

It should be noted here that the encoding representation for the single-frame prediction uses $\tilde{x}_{1:L}$ to extend the context. As for the representation in the attention calculation process, to maintain the element differences among the encoding representations, we still use $x_{1:L}$. Apart from choosing which historical output $y_{t-1}$ is used in the classification process, a choice also needs to be made of appropriate posterior alignment distribution for eqn. (12). Using even more ground-truth label information for this, introduces an additional mismatch between training and inference stages. In order to distinguish between the common exposure bias problem and this additional PAM mismatch, we denote them as “state update” and “posterior choice” mismatch respectively.

**B. POSTERIOR CHOICE AND DIVERGENCE PENALTY**

Unlike a straightforward state update, posterior choice has certain unique attributes. The exposure bias problem in traditional seq2seq depends on the computed value of $y^*_t$. It implies that only if we get a 100% correct sequence during the inference stage can there be no discrepancy at all. Conversely, posterior choice implies selecting the appropriate posterior alignment distribution conditioned on the output. Correct distribution $\alpha^\text{post}_{t}$ is required, rather than that of the ground-truth $y^*_{t-1}$. Overall, the posterior choice has some fault-tolerances for the use of the output knowledge $y_{t-1}$ compared to the state update mismatch.

The calculation of posterior alignment can be regarded as a function of the posterior probability and the prior alignment distribution. According to whether $y_t$ is chosen to be the ground-truth output $y^*_t$ or the model-predicted one $\tilde{y}_t$, we can mark the corresponding posterior alignment as $\alpha^\text{post}_{t}$ or $\tilde{\alpha}^\text{post}_{t}$ respectively.

In the system proposed here, the posterior choice problem is solved by modifying eqn. (12). Specifically, it is to use model predicted output $\tilde{y}_t$ instead of $y^*_t$ to optimize the distribution of posterior alignment, and a penalty loss term introduced to ensure the obtained distribution is similar to the target one. In this way, even though occasional erroneous historical output is unavoidable, its negative influence on posterior alignment is reduced. The loss is found as in eqn. (15), where $\text{Div}$ represents the F-divergence function [37], [38].

$$
\text{Loss}_{\text{Div}} = \lambda \text{Div}(\alpha^\text{post}_{t} || \tilde{\alpha}^\text{post}_{t}) \quad (15)
$$

Penalty weight $\lambda$ needs to be adjusted to an appropriate value to balance the optimisation of the attention score.

**C. STATE UPDATE AND ALTERNATE LEARNING**

In traditional seq2seq models, schedule sampling (SS) [25] tends to be the preferred method to bridge the gap between training and inference, referred to as a state update solution. SS introduces a random probability ($\epsilon_t$) of selecting an inference state during training. Although it is simple to implement and does improve performance, it leads to an accumulation of errors which increases with sequence length. Furthermore, careful adjustment of the hyper-parameters is critical to performance. To solve these deficiencies, we propose a new approach we refer to as an alternate learning strategy (ALS), which integrates one training stage and several auxiliary inference steps at each decoding step.

ALS integrates both training and inference information into the model training process, as illustrated in Fig. 2. Let us explain it by following the decoding process of the $t$-th step. First, the token prediction output ($\tilde{y}_t$) is obtained via the training stage, which uses ground-truth label $y^*_{t-1}$ as its historical context. For the subsequent $M$ steps, from $t + 1$ to $t + M$, auxiliary outputs are generated recursively by using inference labels output from the previous step as the historical source input (eg. $\tilde{y}_t$ acts as the inputs to generate auxiliary output $\tilde{y}_{t+1}$). The prediction outputs in those two phases are marked as $y_{t+1}^{(t)}$, $\tilde{y}_{t+1}^{(t)}$ separately. Since the decoder state $s_{t+M}$ obtained with $\tilde{y}_{t+M-1}$ as input in subsequent $M$ auxiliary inference steps is not passed on to the next time-step, errors will not be accumulated along the token step.
Thus there exists a completely error-free path before the inference stage for every decoding step, i.e. in the Fig. 2 model block, errors could accumulate along the downwards steps, but are not propagated horizontally when moving from left to right. To continue the explanation, consider speech training sample token $T$ as an example. There are $T$ steps to adopt the ground-truth label (the blue area in Fig. 2), which is denoted as the main chain. The calculation process for this main chain is exactly the same as that of the teacher-forcing method [28]. Alongside this, ALS also obtains $M \times T$ auxiliary decoding processes (called the auxiliary grid, shown as the green area) generated by using the model token prediction output. The losses of these two stages, the main chain and the auxiliary grid, correspond to the training and inference losses. The calculation formulae are shown in eqn. (16) and (17), respectively. In the above two equations, a frame-level cross entropy (CE) criterion is adopted as the objective function.

Unlike model parameters updated with the summation of two losses, we update the losses alternately. Continuing our example explanation, the model is firstly updated with the training loss, and then the update operation is repeated with the corresponding inference loss. The updates alternate between the two losses, hence the ‘alternate’ in ALS. Note that we also perform a numerical regularization on the inference loss to ensure the influence of both losses is compatible. The two loss functions are,

\[
\begin{align*}
\text{Loss}_{\text{training}} &= \sum_t L_{CE}(y^*, \hat{y}^{(tr)}_t) \\
\text{Loss}_{\text{inference}} &= \frac{1}{M} \sum_m \sum_t L_{CE}(y^*_m, \hat{y}^{(in)}_{m,t})
\end{align*}
\]

The error-free path before the inference for each step reduces the accumulation of errors caused by historical mistake predictions, leading to more stable model training with fewer super-parameters required. In fact the only parameter in ALS is the number of auxiliary inference steps $M$.

IV. EXPERIMENTAL EVALUATION

To assess the effectiveness of each of these proposed improvements we extensively evaluate a number of system variants on WSJ [26], and finally measure performance on the larger Switchboard corpus [39]. Some additional experiments use the smaller TIMIT [40] dataset, which is better at rapidly analyzing and visualizing issues relating to PAM applied to ASR. Although all databases use English speech, no aspects of the improvements we introduce are language-specific.

For the experiments on TIMIT, We trained on the standard 462 speaker set with all SA utterances removed and used the 50 speaker dev set for early stopping. Final results were obtained on the 24 speaker core test set. The networks were implemented using the Theano library [41]. Decoding was performed using the 61 phoneme set, while scoring was done on the 39 phoneme set. Phone error rate (PER) is used to measure the performance. The other experimental settings for TIMIT, unless explicitly noted below, follow [42]. We further validated the posterior model on two ASR corpora: WSJ and Switchboard. The Switchboard corpus consists of English telephone speech. We use the 300 hours train dataset (LDC97S62), a 90% subset for training, and a small part for cross validation. For evaluation, we compute the word error rate (WER) of HUB5 Eval2000 data set (LDC2002S09), consisting of two subsets: Switchboard (SWB), which is similar in style to the training set, and CallHome (CHE), which contains conversations between friends and family. The WSJ database contains 80 hours of transcribed speech. Here we used the standard configuration of si284 for training, dev93 for validation and we report WER and character error rate (CER) on eval92.

The encoder used for WSJ had 2-layers of convolutions, which down-sample the sequence in time, with $3 \times 3$ filters and 32 channels, followed by 4 layers of bidirectional LSTM (Long Short-Term Memory) with a cell size of 800. While the architecture of Switchboard is the same as WSJ’s except for varying the number of encoder layers to be 6. For input features in Switchboard and WSJ, we use an 80-dimensional log-mel filterbank plus 3 pitch coefficients, with per-speaker mean and variance normalization. The targets of our end-to-end system are a set of 51 characters which contain English letters, numbers, punctuation and special transcribed notations. The other experimental settings we extensively evaluate a number of system variants to assess the effectiveness of each of these proposed improvements. The other experimental settings we extensively evaluate a number of system variants to assess the effectiveness of each of these proposed improvements.
during beam-search, the beam-size was set to 10, 20 and 20 for TIMIT, WSJ and Switchboard respectively. Due to its relatively small training set size, WSJ model accuracy varies slightly between runs and so authors typically repeat four times, reporting the average score. We do the same, also indicating the maximum and minimum scores over those runs for completeness. Switchboard has a much larger training set thus does not require multiple runs.

We evaluate several variant systems here, with structure details as follows:

- **Traditional** is the traditional soft attention-based speech recognition system reported in [8], [46] and [47].
- **PriorAttention**: for fair comparison, we reproduced the above traditional architecture ourselves using ESPnet. This system shares the same training criteria and strategies as the following systems.
- **PAM** is a system in which the posterior attention model is applied directly to the ASR task. The number of parameters in the systems noted below are approximately the same as in this system.
- **PAM\textsubscript{RNN}** is based on the above, but with the inclusion of the optional RNN layer described in eqn. (13).
- **PAM\textsubscript{RNN,Ctx}** is based on PAM\textsubscript{RNN}, but appends one context block (Ctx), as described in eqn. (14), after the encoder.
- **PAM\textsubscript{RNN,Ctx,Loss}** corrects the deficiency of posterior choice in PAM\textsubscript{RNN,Ctx} by incorporating the proposed divergence penalty loss of eqn. (15).
- **PAM\textsubscript{RNN,Ctx,ALS}** incorporates the alternate learning strategy (ALS) into the PAM\textsubscript{RNN,Ctx} system to alleviate the “state update” problem.
- **EPAM**: the final extended PAM system which combines all of the modifications mentioned above.

### A. PRELIMINARY PAM STRUCTURAL EVALUATION

In this experiment, we evaluate the PAM on an ASR task, and compare results to the traditional prior attention approaches [8]. All systems are as described in Section III-A.

| Model          | TIMIT       | WSJ        |
|----------------|-------------|------------|
|                | PER, %      | CER, %     | WER, %     |
| Traditional    | 18.0 [8]    | 4.6 [46]   | 12.9 [46]  |
| PriorAttention | 18.4        | 4.3±0.1    | 12.0±0.2   |
| PAM            | 18.2        | 4.9±0.1    | 13.6±0.3   |
| PAM\textsubscript{RNN} | 18.1        | 4.4±0.1    | 12.4±0.2   |

The results, listed in Table 1, show that, compared to the PriorAttention system, PAM and PAM\textsubscript{RNN} achieve a modest gain on the TIMIT task. The additional update in the decoder state for PAM\textsubscript{RNN} seems to contribute to a performance improvement, and this is more significant on the WSJ task (e.g. 12.4% vs 13.6% WER). We conjecture that the reason for this improvement may be information fusion. The fusion of \(\alpha_{t-1}^{\text{postr}}\) and \(\alpha_t^{\text{prior}}\) in the decoder state may be helpful to exploit complementarity in the prior and posterior alignment vectors. This is because while \(\alpha_{t-1}^{\text{postr}}\) has a more accurate estimation of the previous step, the location information it conveys for the current step may be inferior to the \(\alpha_t^{\text{prior}}\) estimate.

To understand this process more clearly, we take advantage of the frame-level segmentation in TIMIT to visualize the PAM\textsubscript{RNN} system alignment variables in Fig. 3. Firstly, Fig. 3 (a), (b) and (c) present heatmaps of prior alignment \(\alpha_t^{\text{prior}}\), corresponding posterior alignment \(\alpha_t^{\text{postr}}\) and traditional attention, respectively. Compared with the latter, the alignments that the posterior attention model generates correspond well to the spectrogram segments (eg. see the 4-th phoneme “ey”).

Looking further, Fig. 3(d) plots the error prediction in this sentence, and particularly shows errors occurring in the 4-th, 7-th…9-th tokens. Subplot (e) presents the difference between prior and posterior alignment scores, denoted as \(\Delta\alpha_t\). By observing and comparing those two figures, we interestingly see strong position correlation between \(\Delta\alpha_t\) and the error prediction (Fig. 3 (e) and (d)). This may indicate that the correct posterior output information \(\gamma_t^x\) is conducive to the adjustment of alignment vector when the error estimation occurs.

Fig. 3 (f) visualizes the influence of posterior-choice mismatch on the \(\alpha_t^{\text{postr}}\) distribution. The full description and analysis of this will be presented in the next subsection (IV-B), we don’t repeatedly introduce here.

From Fig 3 (g), we can see that the entropy of the posterior alignment score is lower than prior ones (1.76 vs 1.62). Further we analyze the entropy histogram of the \(\alpha_t^{\text{postr}}\), \(\alpha_t^{\text{prior}}\) and traditional attention on the entire TIMIT valid set (Fig. 3 (h) and (i)). From this we find that the posterior mechanism has the lowest entropy value, showing it is more focused (thus has a sharper distribution). The above delta and entropy results jointly indicate that the additional use of posterior information is indeed helpful to the optimization (correction and sharpening) of alignment vectors.

However, looking back at Table 1, we also notice that the performance of the posterior attention systems does not consistently outperform the traditional prior attention systems, especially on WSJ. This is due to the deficiencies noted in Section II, which we will now address in the subsequent systems evaluated below.

| \(k\) | 0 | 1 | 2 | 3 | 4 |
|------|---|---|---|---|---|
| \text{PAM}\textsubscript{RNN,Ctx} | 4.4 | 12.4 | 11.8 | 11.6 | 11.5 |

First, we assess the necessity of enhancing context information, which we do by incorporating the context block (Ctx) in the PAM\textsubscript{RNN,Ctx} system. Since the context filter size, \(2k + 1\), is variable, we will additionally assess several context lengths. Results are also shown in Table 2. As can be...
noticed, \( k = 3 \) yields best results overall. More contextual information does not bring further improvement, which we speculate is due to the increased filter size gathering too much irrelevant temporal information from beyond the char-token’s border in WSJ. However we do note that incorporation of the context block greatly improves performance (from 12.4% to 11.5% WER) not only because of the context information requirements of ASR tasks, but also due to the additional factor of posterior choice mismatch.

### B. REDUCE POSTERIOR INFORMATION MISMATCH

Even though posterior attention can provide better alignment to the traditional attention mechanism, we have discussed above that the posterior information is not optimal in other parts of the system because it introduces mismatches in the “posterior choice” and “state update” mechanisms.

To investigate further, we design an experiment to observe the effect of posterior information on these mechanisms. Using the PAM\(_{\text{RNN}}\) system we create an artificial environment to allow ground-truth information to be used for posterior choice or state update (something that obviously would not be possible in a real world environment) and assess the difference between having ground-truth and mismatched information for each.

In Table 3, we quantify the effect of these two mismatches on ASR performance by constructing three variant PAM\(_{\text{RNN}}\) systems with different status update and posterior choice selections. In the table, \( \text{true} \) means using ground-truth \( y_t^* \) while \( \text{inference} \) means employing the estimate \( \hat{y}_t \). All tests are performed on the TIMIT task. Results in the first row represent the upper bound of the model performance, that is, the model performance without any mismatch between training and inference stages, i.e. with perfect knowledge of ground-truth. It can be seen that both mismatches have an impact on model performance, with the effect of a state-update mismatch being larger than that for a posterior-choice mismatch. Although this test was a contrived experiment (since it had knowledge of ground-truth during inference), it clearly demonstrates the need to alleviate both mismatches. In the following subsections we will discuss this further.

#### 1) POSTERIOR CHOICE AND DIVERGENCE PENALTY LOSS

The effect of “posterior choice” can be seen from the visualization experiment reported above. In Fig. 3 (f), we compared the two kinds of posterior alignment, \( \alpha_t^{\text{post}} \) and \( \bar{\alpha}_t^{\text{prior}} \), obtaining by ground-truth output \( y_t^* \) and the predicted output

---

**FIGURE 3.** Attention visualizations for different attention-based encoder-decoder models; “Tradn” is the traditional seq2seq attention mechanism while “prior” and “postr” are the corresponding prior and posterior alignment mechanisms applied to PAM\(_{\text{RNN}}\). Data for subplots (a) to (g) are taken from analysis of one example TIMIT sentence (MBWM0_SI1934), while histogram subplots (h) and (i) are computed over the entire TIMIT valid set. Each row of the alignment images (a) to (c), (e) and (f) represents frames selected by the attention mechanism to emit a phone symbol, while dashed lines demarcate ground-truth phone segments. Subplots (a) to (c) are the alignment score for the three different type of attention while (e) plots the absolute difference between corresponding posterior and prior alignment scores and (f) does the same between posterior alignment scores obtained from ground-truth and predicted outputs respectively. Those predicted outputs and corresponding ground-truth labels are shown over 14 phonemes in subplot (d). (g) show the entropy value of posterior and prior attention for the example sentence. Histograms (h) and (i) present the attention entropy for the different mechanisms on the whole valid set.

**TABLE 3.** The impact of two kinds of exposure bias problems on the performance of PAM\(_{\text{RNN}}\) on the TIMIT corpus task. ‘true’ and ‘inference’ mean using \( y_t^* \) and \( \hat{y}_t \) respectively for test set evaluation processing.

| State Update | Posterior Choice | Dev PER | Test PER |
|--------------|------------------|---------|----------|
| true         | true             | 15.2    | 16.6     |
| true         | inference        | 15.4    | 16.9     |
| inference    | inference        | 16.8    | 18.4     |
\( \hat{y}_t \) separately. The probability distribution of \( \alpha_{\text{postr}}^{\text{apostr}} \) is obviously different from \( \alpha_{\text{postr}} \) at the error prediction token. This indicates that different \( y_t \) will bring about different alignment distributions in the posterior optimization process. Furthermore, the use of an incorrect posterior information processing alignment distribution in the inference stage will bring about a decrease in alignment accuracy.

To solve this problem, we propose the introduction of a divergence penalty term. We now verify the effect of using different divergence penalties in Table 4. From this it can firstly be seen from the experimental results that Kullback-Leibler (KL) divergence is a better choice which can achieve 0.6% absolute reduction based on the comparable baseline. We suspect this is because the clear target distribution in KL seems to be more important to overall performance, even though Jensen-Shannon (JS) divergence is symmetrical and may be smoother.

**TABLE 4.** The performance of PAM\(_{\text{RNN,Loss}}\) with difference divergence penalty losses for TIMIT. Penalty weight \( \lambda \) is set to 0.2 and 1.0 for KL and JS, respectively.

| Methods            | Dev PER | Test PER |
|--------------------|---------|----------|
| PAM\(_{\text{RNN}}\)  | 16.8    | 18.1     |
| + KL-divergence     | 16.4    | 17.5     |
| + JS-divergence     | 16.5    | 17.8     |

We further verify the effect of divergence penalty on WSJ with results reported in Table 5. Firstly we see that for the PAM\(_{\text{RNN}}\) system, the effect of adding the penalty is very significant, indicating that it is an effective method of improving model performance. However, when context is already incorporated, for example in the PAM\(_{\text{RNN,Ctx}}\) system, the beneficial effect of the penalty reduces. This tells us that the effect of adding context not only provides the model with additional useful information, but also reduces the objective difference between the two distributions. Interestingly, it means that the context block goes part way towards fixing the posterior choice problem.

This conclusion can be verified in Fig 4 which demonstrates the source of that performance gain by plotting KL divergence for train and valid data sets in PAM\(_{\text{RNN}}, \text{PAM}_{\text{RNN,Ctx}}\) both without and with the penalty term. However, despite the benefit of context, the penalty term is still useful for improving performance. That benefit can be visualised more clearly in Fig 4 (b) and (c).

2) ALTERNATE LEARNING AND EXPOSURE BIAS

Unlike with schedule sampling, which should adjust the proportion of inference labels \( \epsilon_i \) and its decay schedule [25], the proposed ALS method only needs the number of auxiliary inference steps \( M \) to be adjusted. This affects the proportion of adaptation “training” and “inference” stages in the model training process, as well as the model convergence rate. Here, we experiment with different values of \( M \) between 1 and 6.

![FIGURE 4. (a) KL-divergence value of training and validation set in the PAM\(_{\text{RNN}}\) system (without penalty). (b) and (c) are the corresponding divergence value in the PAM\(_{\text{RNN}}, \text{PAM}_{\text{RNN,Ctx}}\) when utilizes a divergence-based penalty.](image)

**TABLE 5.** The performance (WERS and CERS) of the PAM\(_{\text{RNN}}\)-PAM\(_{\text{RNN,Ctx}}\) system with and without penalty weight on the WSJ corpus.

| System              | CER   | WER   | CER   | WER   |
|---------------------|-------|-------|-------|-------|
| without penalty     | 4.4±0.1  | 12.4±0.2 | 4.1±0.1 | 11.5±0.2 |
| with penalty        | 4.1±0.1  | 11.5±0.2 | 4.0±0.1 | 11.1±0.3 |

Results are shown in Fig. 5, where we find \( M = 5 \) can achieve 0.6% absolute improvement for WSJ (from 11.5% to 10.9%) which means one training step combined with 5 inference steps (similar to \( \epsilon_i \approx 83.3\% \)) per decoder step as the best choice for ALS. In addition, we note that ALS can make use of a higher “inference” rate in the training process, compared to the \( \epsilon_i \) in schedule sampling. We believe this is because ALS is less prone to error accumulation, thanks to the fact that the model is error-free before the inference stage in each step.
Although ALS does not affect the complexity of the model evaluation phase, it increases training time by requiring $M$ times more decoding steps. Thus, training time per epoch tends to increase linearly with the value of $M$, as shown in Fig 5 (yellow text). Taking $M = 5$ as an example, judged in terms of runtime on a GPU cluster, ALS has around 2.2 times higher training complexity. However, evaluation mode (runtime) complexity is not increased.

### C. THE SUMMARY AND INTEGRATION OF THOSE MODIFICATIONS

In this section, we summarize the impact of the modifications, including context block (Ctx), divergence penalty (Loss) and alternate learning strategy (ALS), on model performance. As shown in Table 6, it seems that PAM$_{RNN}$ benefits most from utilizing the context block, with PAM$_{RNN,Ctx}$ achieving 11.5% on WSJ, an absolute WER improvement of 0.9% over the best baseline architecture (rows 4 and 5). We then study the effects of divergence penalty by fixing the context filtersize to 3. Compared with PAM$_{RNN,Ctx}$ system, PAM$_{RNN,Ctx,Loss}$ can obtain an absolute WER reduction of 0.4% by using divergence penalty (rows 5 and 6). Based on the best teacher forcing trained cross-entropy model, PAM$_{RNN,Ctx,Loss}$, we perform ALS with auxiliary inference step $M = 5$. As show in row 7, the final performance of EPAM achieves 10.6% WER, an absolute improvement of 0.5%. On the other hand, this final proposed EPAM system can yield an 8.6% relative improvement in WER compared with the traditional soft-attention system with SS (rows 3 and 7).

| Methods                  | CER | WER  |
|--------------------------|-----|------|
| Traditional [47]         | 4.6 | 12.9 |
| PriorAttention           | 4.3 ± 0.1 | 12.0 ± 0.2 |
| + SS (0.55)              | 4.1 ± 0.1 | 11.6 ± 0.2 |
| PAM$_{RNN}$              | 4.4 ± 0.1 | 12.4 ± 0.2 |
| PAM$_{RNN,Ctx}$          | 4.1 ± 0.1 | 11.5 ± 0.2 |
| PAM$_{RNN,Ctx,Loss}$     | 4.0 ± 0.1 | 11.1 ± 0.3 |
| EPAM                     | 3.7 ± 0.1 | 10.6 ± 0.3 |

The experiments on WSJ above show that the three modifications can improve system performance. We further verify these conclusions by repeating the same training and evaluation, using the larger Switchboard-300hrs dataset instead. All the results are summarized in Table 7 and, for better comparisons with previous published systems, we also list the results for the Switchboard/CallHome subset of eval2000 without any external language models, respectively. As shown in Table 7, PAM$_{RNN}$ achieves a performance comparable to previous published and traditional systems (17.9% vs 17.9%/17.8% [47]). We observed that two modifications, the context block and divergence penalty, on PAM$_{RNN}$ bring 0.8% absolute improvement, from 17.9% to 17.1%, on the eval2000 dataset. After considering the ALS update strategy for tackling the exposure bias problem, our best model, EPAM, achieves 16.3% WER, which is a 1.0% absolute improvement over the scheduled sampling trained prior attention model (row 3 and 7 in Table 7).

### V. CONCLUSION

Attention mechanisms have shown their worth in improving ASR performance over the past several years, by endowing the classifier with an effective ability to focus on specific regions of the input features. However current systems form an attention vector from contextual input information, which is used to condition output posteriors. This paper explores the hypothesis that posterior information, from the encoder output itself, could be valuable when forming an attention mechanism in encoder-decoder ASR networks — in other words that attention may be improved through knowledge of the output as well as the input. Posterior information has been used in another seq2seq modeling task, namely machine translation, where the posterior attention modeling (PAM) architecture was introduced. PAM forms an attention vector from posterior information and has been shown to improve performance in that research field.

However ASR and machine translation differ in several aspects. This paper first evaluates PAM for ASR, and finds that performance is poor due to those differences. We propose a number of system-level modifications to mitigate the disadvantages brought about by use of posterior information, particularly the posterior choice mismatch, as well as to tackle the common exposure bias problem (which affects many systems, not just those described here). This paper separately evaluates the benefit of each novel contribution using standard ASR tasks from TIMIT and WSJ. The final system is shown to achieve outstanding performance on the WSJ task, of 10.6% WER, compared with 12.0% for a soft prior attention system and 11.6% for the system using schedule sampling method. Further experiments on Switchboard-300hrs dataset show that EPAM achieves 16.3% WER on the eval2000 test set, an absolute WER improvement of 1.0% over the traditional scheduled sampling baseline.

In future, we aim to investigate methods of combining both prior and posterior mechanisms, by allowing an adaptive data-driven model to optimally select either a prior or a posterior alignment score.
[46] M. Karthick Baskar, L. Burget, S. Watanabe, M. Karafiát, T. Hori, and J. H. Černocký, “Promising accurate prefix boosting for sequence-to-sequence ASR,” 2018, arXiv:1811.02770. [Online]. Available: http://arxiv.org/abs/1811.02770
[47] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, “Improving attention based sequence-to-sequence models for end-to-end English conversational speech recognition,” in Proc. Interspeech, Sep. 2018, pp. 761–765.

JIAN TANG received the B.S. degree from the School of Computer Science and Information Engineering, Hefei University of Technology (HFUT), in 2014. He is currently pursuing the Ph.D. degree with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China (USTC). His current research interests include deep learning for speech recognition and acoustic modeling.

JUNFENG HOU received the B.S. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), in 2013. He is currently pursuing the Ph.D. degree with USTC. From October 2014 to April 2015, he was an Intern at iFlytek. His current research interests include deep learning for language model and speech recognition.

YAN SONG received the B.Sc. degree in electronic engineering from the University of Electronic Science and Technology of China, in 1994, and the M.Sc. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 1997 and 2006, respectively. He is currently an Associate Professor with the National Engineering Laboratory for Speech and Language Information Processing. He has been a Faculty Member with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, since 2000. His research interests include multimedia information processing, automatic language identification, speaker diarization, and image classification.

LI-RONG DAI (Member, IEEE) was born in China, in 1962. He received the B.S. degree in electrical engineering from Xidian University, Xi’an, China, in 1983, and the M.S. degree from the Hefei University of Technology, Hefei, China, in 1986, and the Ph.D. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, in 1997. He joined USTC, in 1993, where he is currently a Professor with the School of Information Science and Technology. His research interests include speech synthesis, speaker and language recognition, speech recognition, digital signal processing, voice search technology, machine learning, and pattern recognition. He has published more than 50 articles in these areas.

IAN MCLoughlin (Senior Member, IEEE) received the Ph.D. degree in electronic and electrical engineering from the University of Birmingham, U.K., in 1997. He worked for over ten years in the research and development industry and about 15 years in academia, on three continents. He is currently a Professor with the Singapore Institute of Technology. He has written many articles and several patents on speech analysis and communications. He is the author of four books on speech processing and embedded computation. He is a Fellow of the IET, a Chartered Engineer, a recipient of the Chinese Academy of Sciences President’s International Fellowship Award and the Hundred Talent Program funding from Anhui, China.

***