Exploring Energy-based Language Models with Different Architectures and Training Methods for Speech Recognition

Hong Liu[1,†], *Zhaobiao Lv*[2,†], *Zhijian Ou*[*,1], *Wenbo Zhao*[2], *Qing Xiao*[2]

[1]Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, Beijing, China
[2]China Unicom (Guangdong) Industrial Internet Co., Ltd.

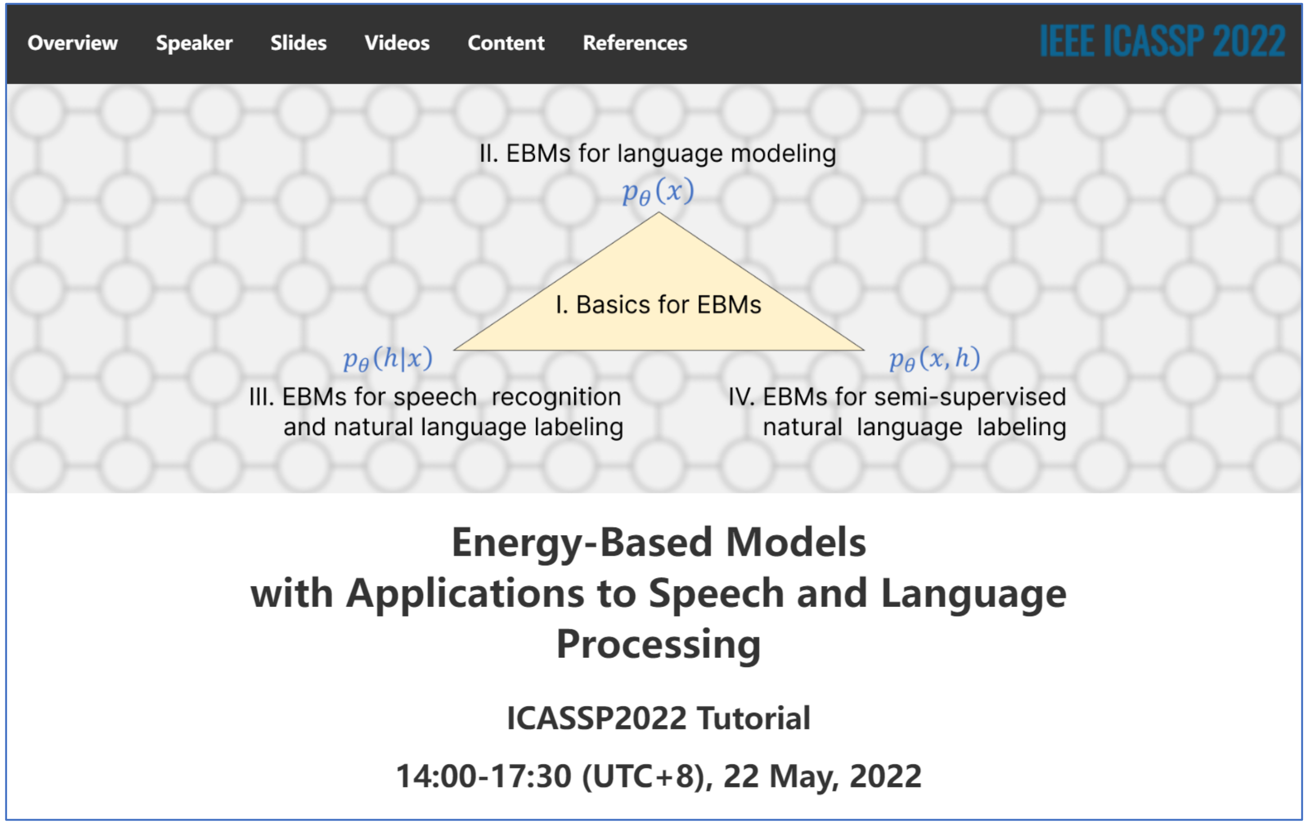
liuhong21@mails.tsinghua.edu.cn, lvzb7@chinaunicom.cn, ozj@tsinghua.edu.cn,
{zhaowb19, xiaoq17}@chinaunicom.cn

Outline

- **Motivation**
- **Models & Methods**
- **Experiments**
- **Conclusion**

Energy-based model (EBM)/random fields/undirected graphical models

https://simons.berkeley.edu/talks/ilya-sutskever-openai-2023-08-14 (Workshop on Large Language Models and Transformers)

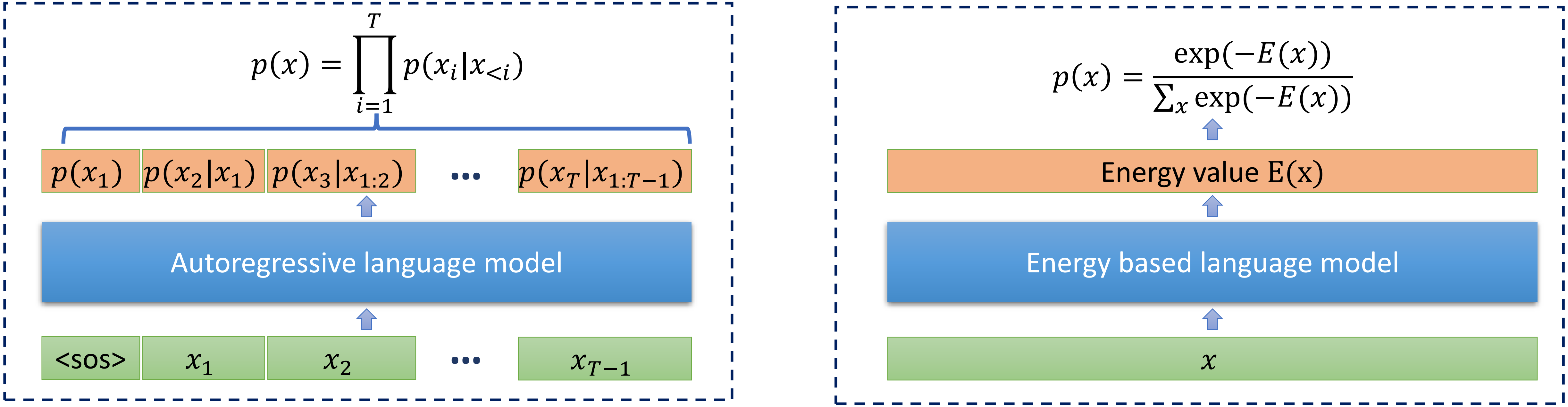


http://oa.ee.tsinghua.edu.cn/~ouzhijian/ICASSP2022/index.html (ICASSP 2022 Tutorial)

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(-E_\theta(x))$$

Autoregressive LM (ALM) vs Energy based LM (ELM)

- Given a sentence $x = \{x_1, x_2, \dots, x_T\}$



- What's the advantage of Energy based language model (ELM)?
  - We can define very flexible energy functions by utilizing neural networks of various architectures
  - ELMs (globally normalized) overcome the exposure bias [1] and label bias [2] suffered by locally-normalized models

[1] S. Wiseman and A. M. Rush, "Sequence-to-sequence learning as beam-search optimization," *EMNLP*, 2016.
[2] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *International conference on Machine learning (ICML)*, 2001.

Motivation

- Applications of ELMs
  - ➢ Computation of sentence likelihoods (up to a constant) [4, 5, 10, 11, 12, 13], text generation [14], language model pretraining [15], calibrated natural language understanding [16], and so on.
- For rescoring in ASR, previous ELMs [3, 4] outperform ALMs with similar model sizes, but they use old-fashioned CNN or LSTM
  - ➢ Recent progress in Transformer and large pretrained models such as BERT and GPT opens new possibility to further advancing ELMs
  - ➢ Explore different architectures of energy functions and different training methods

[3] B. Wang and Z. Ou, "Language modeling with neural trans dimensional random fields," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.
[4] B. Wang and Z. Ou, "Learning neural trans-dimensional random field language models with noise-contrastive estimation," *ICASSP*, 2018.
[5] B. Wang, Z. Ou, and Z. Tan, "Learning trans-dimensional random fields with applications to language modeling," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2018.

Outline

- Motivation
- Models & Methods
- Experiments
- Conclusion

Definition and Notation

- Two different forms of ELMs:

  - (Generally) Globally normalized ELM (GN-ELM)

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{\sum_{x'} \exp(-E_\theta(x'))} = \frac{\exp(-E_\theta(x))}{Z(\theta)}$$

x: sentence (i.e. a token sequence)
θ: model parameters
$E_\theta(x)$: energy of sentence x
$p_\theta(x)$: the probability of x
$Z(\theta)$: normalizing constant

  - Trans-dimensional random field language model [5] (TRF-LM)

$$p_\theta(x) = \pi_{|x|} \frac{\exp(-E_\theta(x))}{\sum_{|x'|=|x|} \exp(-E_\theta(x'))} = \pi_{|x|} \frac{\exp(-E_\theta(x))}{Z_{|x|}(\theta)}$$

$|x|$: token length of x
$\pi_{|x|}$: prior probability of length $|x|$
$Z_{|x|}(\theta)$: normalizing constant at length $|x|$

[5] B. Wang, Z. Ou, and Z. Tan, "Trans-dimensional random fields for language modeling," *ACL (Long Papers)*, 2015.

Definition and Notation

No matter for TRF-LMs or GN-ELMs, one is generally free to choose the energy function in ELMs, as long as it assigns a scalar energy to every sentence.

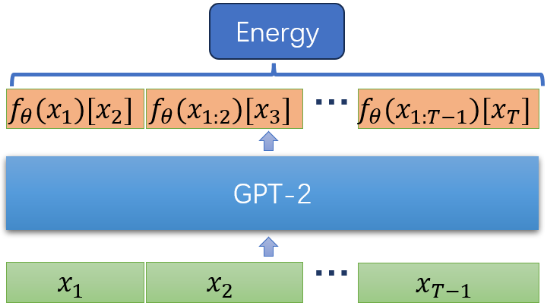
Architectures of Energy Functions

- SumTargetLogit
- Hidden2Scalar
- SumMaskedLogit
- SumTokenLogit

Architectures of Energy Functions

- **SumTargetLogit** [6]: adapted from autoregressive language model (GPT-2), this energy function sums the logits corresponding to the target token (next token) at each position

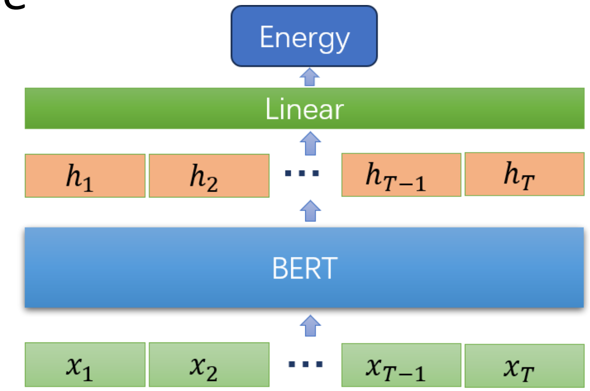
$$E_\theta(x) = -\sum_{i=1}^{|x|} f_\theta(x_{1:i-1})[x_i]$$



x_i: the i-th token (also token id) in x
$f_\theta(x_{1:i-1})$: the output logits for predicting the i-th token

- **Hidden2Scalar** [7]: adapted from bi-directional text encoder (BERT), the hidden states of the sentence is projected to scalar space

$$E_\theta(x) = -\mathrm{Linear}\left(\sum_{i=1}^{|x|} \mathrm{enc}_\theta(x)[i]\right)$$



enc_θ: the bi-directional text encoder
$Linear$: a trainable linear layer whose output is a scalar

[6] B. Wang and Z. Ou, "Improved training of neural trans dimensional random field language models with dynamic noise contrastive estimation," in *2018 IEEE SLT.*
[7] Y. Deng, A. Bakhtin, M. Ott, A. Szlam, and M. Ranzato, "Residual energy-based models for text generation," arXiv preprint arXiv:2004.11714, 2020.

Architectures of Energy Functions

- **SumMaskedLogit** [8]: Based on masked language model (MLM), this energy function sums the output logit of masked tokens

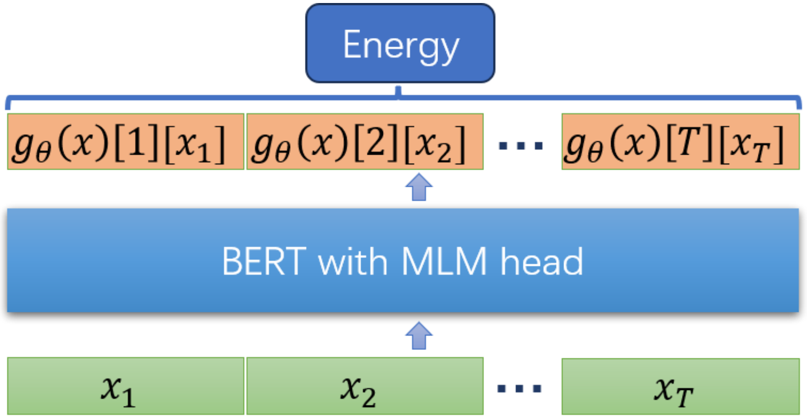
It requires $|x|$ forward pass, very time-consuming!!

$$E_\theta(x) = -\sum_{i=1}^{|x|} g_\theta(\mathrm{MASK}(x,i))[i][x_i]$$

$g_\theta(MASK(x,i))$: the output logits obtained by masking the i-th token and sending the masked sequence into the MLM.

- **SumTokenLogit:** An improvement of SumMaskedLogit. We omit the masking step and feeding x directly to the MLM, so that the logits at all positions can be calculated in parallel.

$$E_\theta(x) = -\sum_{i=1}^{|x|} g_\theta(x)[i][x_i]$$



[8] A. Wang and K. Cho, "BERT has a mouth, and it must speak: BERT as a Markov random field language model," *Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 2019.

Training methods for ELMs

- The normalizing constant $Z(\theta)$ is **intractable** !!

- There are two main classes of training methods for ELMs
- Maximum likelihood estimate (MLE)
  - Metropolis Independence Sampling (MIS)
  - Importance Sampling (IS)
- Noise contrastive estimate (NCE)
  - Standard NCE
  - Dynamic NCE

Training Method: Noise Contrastive Estimate (NCE)

- NCE [9] optimizes the ELM by learning from discrimination between data samples and noise samples.

$$\mathcal{J}_{\text{NCE}}(\theta) = \mathop{\mathbb{E}}_{x \sim p_{\text{data}}} \log \frac{\hat{p_\theta}(x)}{\hat{p_\theta}(x)+\nu q_\phi(x)} + \nu \mathop{\mathbb{E}}_{x \sim q_\phi} \log \frac{\nu q_\phi(x)}{\hat{p_\theta}(x)+\nu q_\phi(x)}$$

q_ϕ: the noise model which is able to generate noisy sentences.
$\hat{p}_\theta(x)$: the **unnormalized** probability $\exp(-E_\theta(x))$
ν: the ratio between the noise prior and the data prior

- The implementation of the noise model q_ϕ: a fine-tuned GPT-2

[9] M. Gutmann and A. Hyvarinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," *Conference on artificial intelligence and statistics (AISTAT)*, 2010.

Training Method: Dynamic Noise Contrastive Estimate (NCE)

- The noise distribution q_ϕ(x) should not be too far or too close to the data distribution.

Too easy to distinguish

Too hard to distinguish

- DNCE [10] optimizes the noise model together with the energy model (teacher-forcing the noise model over the training data)
  - The binary classification task in NCE will gradually become difficult.

$$\mathcal{J}_{\text{DNCE}}(\theta, \phi) = \mathcal{J}_{\text{NCE}}(\theta) + \mathbb{E}_{x \sim p_{\text{data}}} \log q_\phi(x)$$

[10] B. Wang and Z. Ou, "Improved training of neural trans dimensional random field language models with dynamic noise contrastive estimation," *IEEE Spoken Language Technology Workshop (SLT)*, 2018.

Training Method: Maximum Likelihood Estimate (MLE)

- MLE maximizes the likelihood $p_\theta(x)$ over training data
- The gradient of log-likelihood requires Monte Carlo sampling from the energy-based model $p_\theta(x)$

$$\frac{\partial \mathcal{J}_{\text{MLE}}(\theta)}{\partial \theta} = -\mathbb{E}_{x \sim p_{\text{data}}}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] + \mathbb{E}_{x \sim p_\theta}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right]$$

- To sample from a un-normalized distribution $p_\theta(x)$
  - ➢ MCMC methods such as Metropolis-Hasting [11]
  - ➢ Importance sampling [11]

Both require a proposal distribution q_ϕ(x) .
We implement it by a fine-tuned GPT2,
the same as the noise model in NCE.

[11] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer, 2001.

Training Method: MLE with two different sampling methods

- **Metropolis Independence Sampling (MIS):** a special case of Metropolis-Hasting
- Obtain a Markov chain of $p_\theta(x)$ through multi-step of accepting/rejecting proposed samples.

Algorithm 1 Metropolis Independence Sampling in ELM.

Input: A target distribution p_θ, a proposal distribution q_ϕ, iteration number T.

Randomly initialize $x^{(0)}$;

for t =1 to T **do**

Generate x' from the proposal q_ϕ;

Accept $x^{(t)} = x'$ with probability

$\boxed{\min\{1, \frac{p_\theta(x')q_\phi(x^{(t-1)})}{p_\theta(x^{(t-1)})q_\phi(x')}\}}$, otherwise set $x^{(t)} = x^{(t-1)}$;

end for

Return: $\{x^{(1)}, ..., x^{(T)}\}$

- **Importance sampling (IS)**: calculates the weighted sum of energy gradients of the proposed samples
- The second term in MLE gradient:

$$\mathbb{E}_{x\sim p_\theta}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] \approx \frac{\sum_{i=1}^N w(x_i')\frac{\partial E_\theta(x_i')}{\partial \theta}}{\sum_{i=1}^N w(x_i')}$$

- The weight $w(x_i') = \frac{p_\theta(x_i')}{q_\phi(x_i')}$

Outline

- **Motivation**
- **Models & Methods**
- **Experiments**
- **Conclusion**

Experiment Settings

- Datasets

Dataset	Language	Size
AISHELL-1 [12]	mandarin	178 hours
WenetSpeech [13]	mandarin	1000+ hours

- The ASR n-best lists are obtained from a RNN-T [14] model, where the encoder is a Conformer [15] of 92M parameters
- The backbone of energy model is either BERT or GPT-2
- The noise/proposal model is fine-tuned from GPT-2

[12] H. Bu, et al, "Aishell-1: An open source mandarin speech corpus and a speech recognition base line," *O-COCOSDA*, 2017.
[13] B. Zhang, et al, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," ICASSP, 2022.
[14] A. Graves, "Sequence transduction with recurrent neural networks," arXiv preprint arXiv:1211.3711, 2012.
[15] A. Gulati, et al., "Conformer: Convolution augmented transformer for speech recognition," arXiv 2020.

Results on AISHELL-1

- GN-ELM with Hidden2Scalar + DNCE achieves results competitive with the finetuned GPT2
- DNCE outperforms NCE
- GN-ELM and TRF-LM perform closely to each other
- MLE underperforms NCE/DNCE

Table 1: *Rescoring results on AISHELL-1. CER_1 and CER_2 denote the Character Error Rate (CER) in in-domain test and cross-domain test respectively.*

Method	Architecture	CER_1	CER_2
No LM		4.76	5.14
5-gram LM		4.67	4.40
Pretrained GPT2		3.22	3.66
Pretrained BERT (PLL)		3.29	3.66
Finetuned GPT2		3.11	3.33
Finetuned BERT (PLL)		3.12	3.47
NCE (GN-ELM)	SumTargetLogit	3.32	3.39
	Hidden2Scalar	3.20	3.36
	SumTokenLogit	3.27	3.43
DNCE (GN-ELM)	SumTargetLogit	3.25	3.40
	Hidden2Scalar	3.11	3.34
	SumTokenLogit	3.15	3.43
DNCE (TRF-LM)	SumTargetLogit	3.11	3.44
	Hidden2Scalar	3.13	3.39
	SumTokenLogit	3.21	3.47
MLE-IS (GN-ELM)	SumTargetLogit	3.42	3.61
	Hidden2Scalar	3.36	3.48
	SumTokenLogit	3.26	3.41
MLE-MIS (GN-ELM)	SumTargetLogit	3.35	3.59
	Hidden2Scalar	3.26	3.39
	SumTokenLogit	3.25	3.49

Results on WenetSpeech

- GN-ELM with SumTokenLogit + DNCE outperforms the finetuned GPT-2 and finetuned BERT!

- Is the improvement significant?

	p-value (CER1)	p-value (CER2)
SumTokenLogit +DNCE (GN-ELM) vs Finetuned GPT2	0.577	0.015
SumTokenLogit +DNCE (GN-ELM) vs Finetuned BERT	1e-7	0.008

p-value<0.05, significantly improved!

- The advantage of ELMs are more obvious in large dataset!

Table 2: *Rescoring results on WenetSpeech. CER_1 and CER_2 denote the CER in two test sets, TEST-NET and TEST-MEETING, respectively.*

Method	Architecture	CER_1	CER_2
No LM		9.69	17.91
Pretrained GPT2		9.10	15.75
Pretrained BERT (PLL)		9.07	15.69
Finetuned GPT2		8.82	15.52
Finetuned BERT (PLL)		8.96	15.55
DNCE (GN-ELM)	SumTargetLogit	9.03	16.02
	Hidden2Scalar	8.98	15.69
	SumTokenLogit	8.81	15.47
DNCE (TRF-LM)	SumTargetLogit	8.97	15.77
	Hidden2Scalar	8.95	15.67
	SumTokenLogit	9.00	15.65

Confidence Estimate Performance

- Large neural networks are often mis-calibrated (over-confident)
- EBMs are trained with better calibration [33] - its confidence is a good estimate of the actual probability the prediction is correct
- We change the confidence threshold and calculate the precision and recall in test set
- GN-ELM on WenetSpeech achieves a higher AUC, illustrating a better confidence estimate performance than the finetuned GPT2.

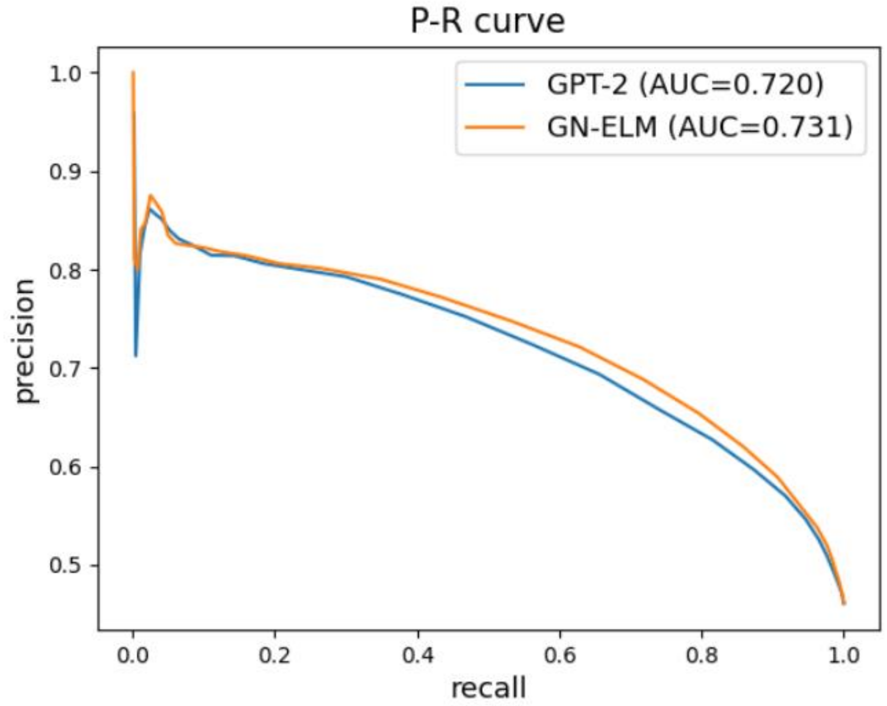


Figure 1: *The confidence estimate performace of the finetuned GPT2 and the best ELM on the TEST-NET of WenetSpeech.*

[33] W. Grathwohl, et al, "Your classifier is secretly an energy based model and you should treat it like one," ICLR 2020.

Outline

- **Motivation**
- **Models & Methods**
- **Experiments**
- **Conclusion**

Conclusion

- An exploration of energy-based language models (ELMs) with different architectures and training methods for rescoring in ASR
- Promising results of ELMs in outperforming locally normalized LM in applications of rescoring and confidence estimate.

Architectures
- SumTargetLogit
- Hidden2Scalar
- SumMaskedLogit
- SumTokenLogit

Training Method
- NCE
- DNCE
- MLE with MIS
- MLE with IS

Application
- LM rescoring in ASR
- Confidence estimate in ASR

Thanks!

Code released at
https://github.com/thu-spmi/CAT/blob/master/docs/energy-based_LM_training.md