Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Gonçalo M. Correia

Jury: André Martins, Mário Figueiredo, Ivan Titov, Wilker Aziz, Isabel Trancoso

Deep learning successes

Deep learning successes

- Subset of machine learning that uses **neural networks**

Deep learning successes

- Subset of machine learning that uses **neural networks**
- Powerful tool for learning representations of any data

Deep learning successes

- Subset of machine learning that uses **neural networks**
- Powerful tool for learning representations of any data
- Remarkable results

Deep learning successes

- Subset of machine learning t
- Powerful tool for learning re
- Remarkable results


A robot wrote this entire article. Are you scared yet, human?

GPT-3

The Guardian

News website of the year

Deep learning successes

- Subs
- Pow
- Rem

WIRED SUBSCRIBE

WILL KNIGHT BUSINESS MAR 29, 2021 7:00 AM

This AI Can Generate Convincing Text—and Anyone Can Use It

The makers of Eleuther hope it will be an open source alternative to GPT-3, the well-known language program from OpenAI.

A robot wrote this entire article. Are you scared yet, human?

GPT-3

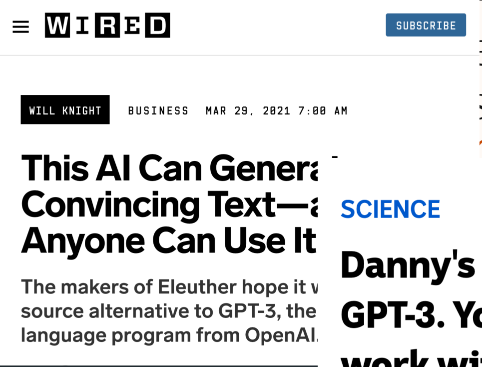
The Guardian
News website of the year

Deep learning successes

- Sub
- Pow
- Rem


A robot wrote this entire article. Are you scared yet, human?

GPT-3

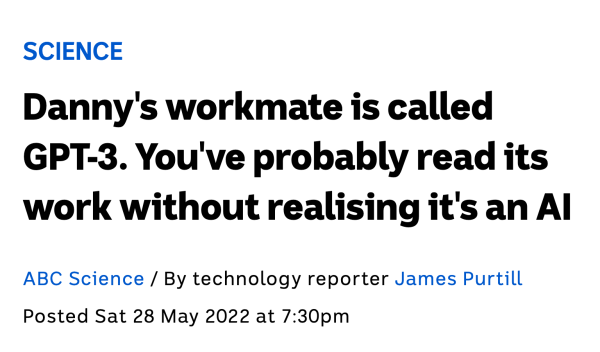

WIRED

SUBSCRIBE

WILL KNIGHT BUSINESS MAR 29, 2021 7:00 AM

This AI Can Genera
Convincing Text—a
Anyone Can Use It

The makers of Eleuther hope it w
source alternative to GPT-3, the
language program from OpenAI.


SCIENCE

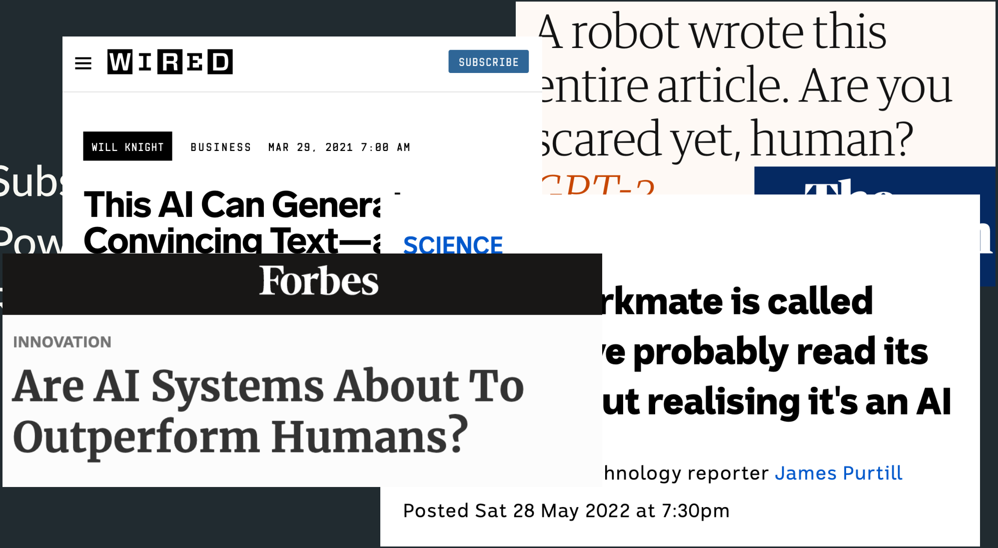
Danny's workmate is called GPT-3. You've probably read its work without realising it's an AI

ABC Science / By technology reporter James Purtill

Posted Sat 28 May 2022 at 7:30pm

Deep learning successes

- Subs
- Pow
- R


WIRED SUBSCRIBE

WILL KNIGHT BUSINESS MAR 29, 2021 7:00 AM

This AI Can Genera
Convincing Text—a

A robot wrote this
entire article. Are you
scared yet, human?

SCIENCE

Forbes

INNOVATION

Are AI Systems About To
Outperform Humans?

rkmate is called
ve probably read its
ut realising it's an AI

hnology reporter James Purtill

Posted Sat 28 May 2022 at 7:30pm

Deep learning successes

Artificial intelligence beats eight world champions at bridge

Victory marks milestone for AI as bridge requires more human skills than other strategy games

robot wrote this
ntire article. Are you
cared yet, human?

PT-2

INNOVATION

Are AI Systems About To Outperform Humans?

rkmate is called
e probably read its
ut realising it's an AI

hnology reporter James Purtill

Posted Sat 28 May 2022 at 7:30pm

Deep learning successes

robot wrote this

Artificial intelligence beats eight world champions at bridge

Victory marks milestone for AI … bridge requires more human sk… than other strategy games

INNOVATION

Are AI Systems About T… Outperform Humans?

Posted Sa…

AI 'outperforms' doctors diagnosing breast cancer

Fergus Walsh
Medical correspondent
@BBCFergusWalsh

Deep learning limitations and drawbacks

Deep learning limitations and drawbacks

- Requires a lot of data

Deep learning limitations and drawbacks

- Requires a lot of data
- Hard to understand and interpret reasons behind decisions

Deep learning limitations and drawbacks

- Requires a lot of data
- Hard to understand and interpret reasons behind decisions
- Requires a lot of computation

Deep learning limitations and drawbacks

WIRED

SUBSCRIBE

AI Can Do Great Things—if It Doesn't Burn the Planet

The computing power required for AI landmarks, such as recognizing images and defeating humans at Go, increased 300,000-fold from 2012 to 2018.

- Requires a lot of data
- Hard to understand and inte … sions
- Requires a lot of computation

Deep learning limitations and drawbacks

- Req
- Har sions
- Req

Forbes

AI

Overcoming AI's Transparency Paradox

WIRED

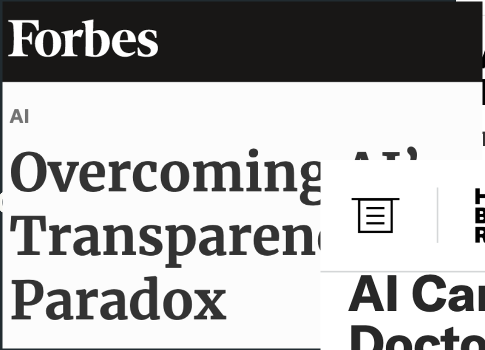
SUBSCRIBE

AI Can Do Great Things—if It Doesn't Burn the Planet

The computing power required for AI landmarks, such as recognizing images and defeating humans at Go, increased 300,000-fold from 2012 to 2018.

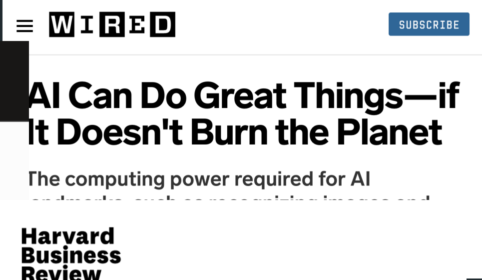
Deep learning limitations and drawbacks

- Req
- Har sions
- Req


Forbes

AI

Overcoming AI's Transparen Paradox


WIRED

SUBSCRIBE

AI Can Do Great Things—if It Doesn't Burn the Planet

The computing power required for AI


Harvard Business Review

AI Can Outperform Doctors. So Why Don't Patients Trust It?

by Chiara Longoni and Carey K. Morewedge

Key concepts of this thesis

Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Key concepts of this thesis

Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Key concepts of this thesis

Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Key concepts of this thesis

Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Key concepts of this thesis

Learnable Sparsity and Weak Supervision for Data-efficient, Transparent, and Compact Neural Models

Key concepts of this thesis

Learnable Sparsity and Weak Supervision

for Data-efficient, Transparent, and Compact

Neural Models

Published work of this thesis

(Correia and Martins, 2019)

Published work of this thesis

- Automatic Post-Editing using weak supervision for data-efficiency (ACL)

(Correia, Niculae, and Martins, 2019)

Published work of this thesis

- Automatic Post-Editing using weak supervision for data-efficiency (ACL)
- Letting transformer learn sparsity of its attentions for transparency (EMNLP)

(Correia, Niculae, Aziz, and Martins, 2020)

Published work of this thesis

- Automatic Post-Editing using weak supervision for data-efficiency (ACL)
- Letting transformer learn sparsity of its attentions for transparency (EMNLP)
- General strategy for efficiently training discrete latent variable models, to have compactness (NeurIPS)

Table of Contents

A bit of context on transformers

What if... Attention is all you need?

A bit of context on transformers

What if... Attention is all you need?

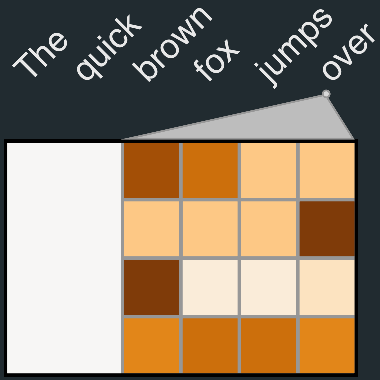
Key idea: Let's mainly use attention mechanisms!

A bit of context on transformers

What if... Attention is all you need?

Key idea: Let's mainly use attention mechanisms!

- Do attention with multiple heads (i.e. attention mechanisms in parallel)



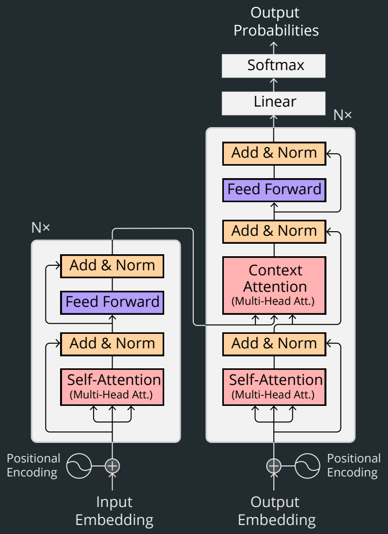
(Vaswani et al., 2017)

A bit of context on transformers

What if... Attention is all you need?

Key idea: Let's mainly use attention mechanisms!

- Do attention with multiple heads (i.e. attention mechanisms in parallel)
- ... and do it through several layers



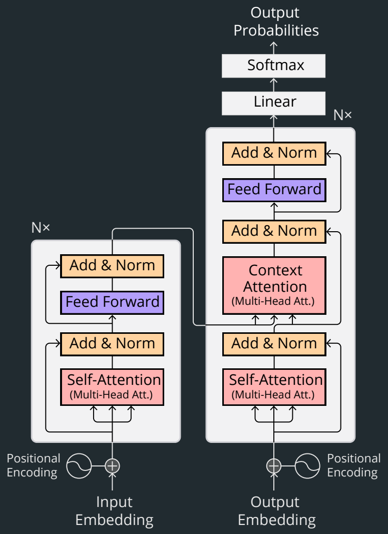
(Devlin et al., 2019)

A bit of context on transformers

What if... Attention is all you need?

Key idea: Let's mainly use attention mechanisms!

- Do attention with multiple heads (i.e. attention mechanisms in parallel)
- ... and do it through several layers
- Inspiration for big general-purpose models like BERT and GPT-3!



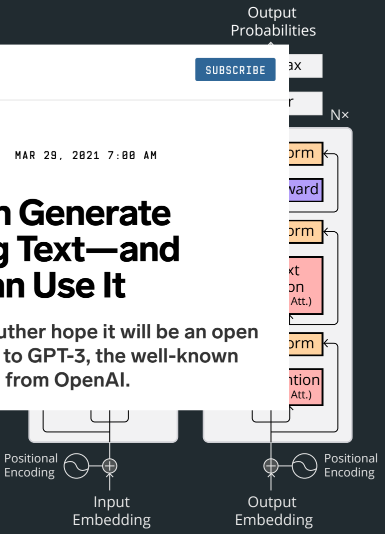
(Devlin et al., 2019)

A bit of context on transformers

What if... Attention is all you need?

Key idea: Let's mainly use attention mech

- Do attention with multiple heads (i.e. mechanisms in parallel)
- ... and do it through several layers
- Inspiration for big general-purpose models like BERT and GPT-3!

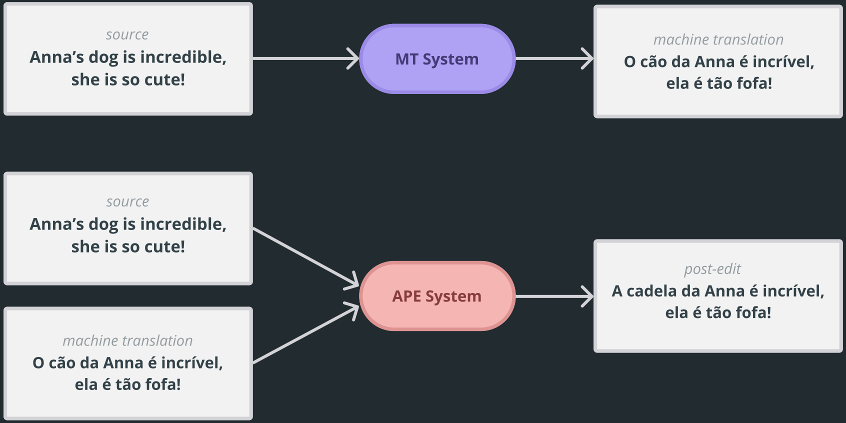


WIRED
SUBSCRIBE
WILL KNIGHT BUSINESS MAR 29, 2021 7:00 AM

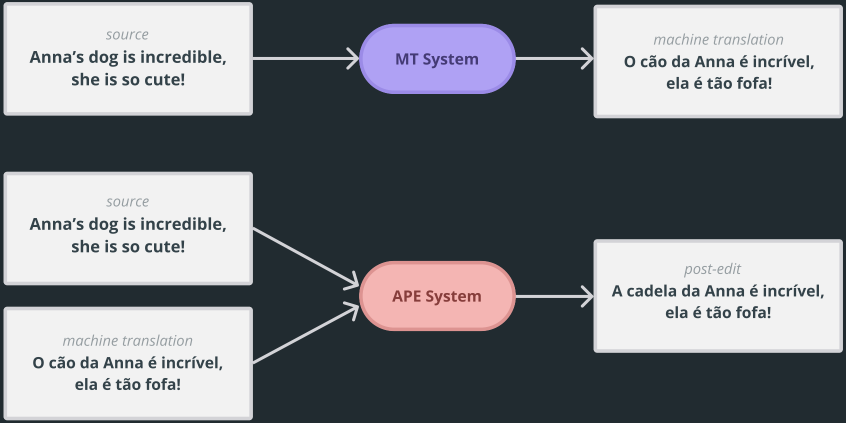
This AI Can Generate Convincing Text—and Anyone Can Use It

The makers of Eleuther hope it will be an open source alternative to GPT-3, the well-known language program from OpenAI.

What is APE?

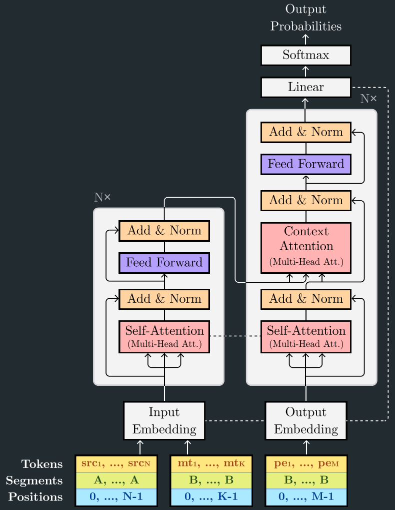

source
Anna's dog is incredible, she is so cute!
MT System
machine translation
O cão da Anna é incrível, ela é tão fofa!
source
Anna's dog is incredible, she is so cute!
machine translation
O cão da Anna é incrível, ela é tão fofa!
APE System
post-edit
A cadela da Anna é incrível, ela é tão fofa!

What is APE?



Challenge: APE data is very scarce! Need to create artificial data.

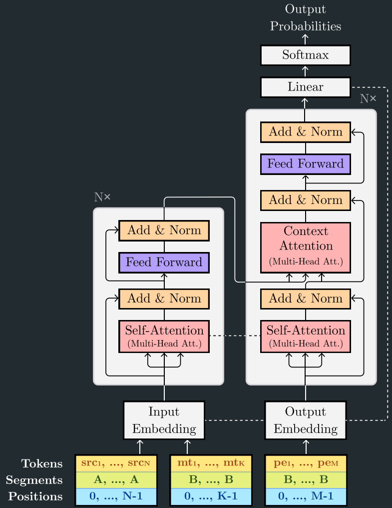
BERT for APE



Output Probabilities
Softmax
Linear
N×
Add & Norm
Feed Forward
Add & Norm
Context Attention (Multi-Head Att.)
Add & Norm
Self-Attention (Multi-Head Att.)
N×
Add & Norm
Feed Forward
Add & Norm
Self-Attention (Multi-Head Att.)
Input Embedding
Output Embedding
Tokens
Segments
Positions
src1, ..., srcN
A, ..., A
0, ..., N-1
mt1, ..., mtK
B, ..., B
0, ..., K-1
pe1, ..., peM
B, ..., B
0, ..., M-1

BERT for APE

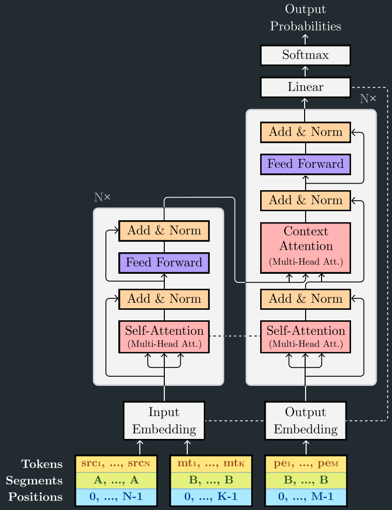
Key idea: Use BERT to do APE



BERT for APE

Key idea: Use BERT to do APE

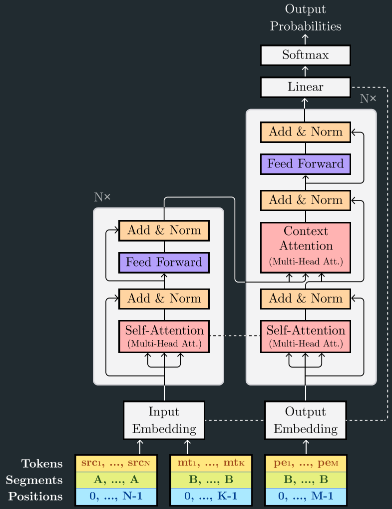
- Prior to this work, BERT was mainly used for simple classification tasks



BERT for APE

Key idea: Use BERT to do APE

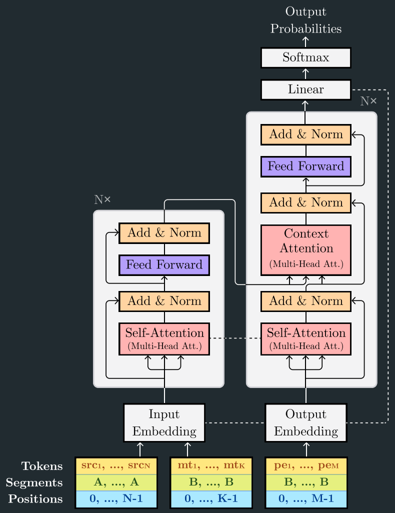
- Prior to this work, BERT was mainly used for simple classification tasks
- We introduced an effective method to use BERT in a generation task (APE)



BERT for APE

Key idea: Use BERT to do APE

- Prior to this work, BERT was mainly used for simple classification tasks
- We introduced an effective method to use BERT in a generation task (APE)
- Smart parameter sharing between encoder and decoder



Key results

model (data size)	TER↓	BLEU↑
mt baseline	24.48	62.49

Key results

model (data size)	TER↓	BLEU↑
mt baseline	24.48	62.49
dual-source transformer (8M)	18.10	71.72

Key results

model (data size)	TER↓	BLEU↑
mt baseline	24.48	62.49
dual-source transformer (8M)	18.10	71.72
dual-source transformer (23K)	27.73	59.78

Key results

model (data size)	TER↓	BLEU↑
mt baseline	24.48	62.49
dual-source transformer (8M)	18.10	71.72
dual-source transformer (23K)	27.73	59.78
ours (23K)	19.03	70.66

Key results

model (data size)	TER↓	BLEU↑
mt baseline	24.48	62.49
dual-source transformer (8M)	18.10	71.72
dual-source transformer (23K)	27.73	59.78
ours (23K)	19.03	70.66
ours (8M)	17.26	73.42

Conclusions and impact

Conclusions and impact

- One of pioneers in using pre-trained transformer encoders for a generation task

Conclusions and impact

- One of pioneers in using pre-trained transformer encoders for a generation task
- Massive improvement in low-resource scenario (data-efficiency)

(Lee et al., 2020)

Conclusions and impact

- One of pioneers in using pre-trained transformer encoders for a generation task
- Massive improvement in low-resource scenario (data-efficiency)
- Steered SOTA of APE towards using weak supervision through pre-trained models

(Kodama et al., 2020)

Conclusions and impact

- One of pioneers in using pre-trained transformer encoders for a generation task
- Massive improvement in low-resource scenario (data-efficiency)
- Steered SOTA of APE towards using weak supervision through pre-trained models
- Inspired other works that use scarse data (*e.g.*, dialogue with metadata)

Table of Contents

Getting to know attention heads better

Attention heads may aid visualization but they are completely dense.

Getting to know attention heads better

Attention heads may aid visualization but they are completely dense.

Our solution is to bet on sparsity:

- for interpretability
- for discovering linguistic structure
- for efficiency

Getting to know attention heads better

Attention heads may aid visualization but they are completely dense.

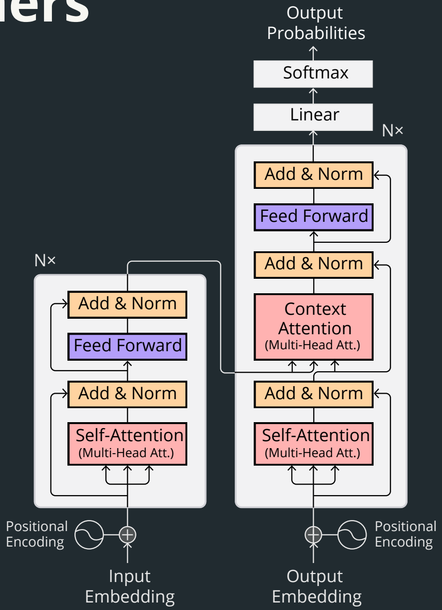
Our solution is to bet on sparsity:

- for interpretability
- for discovering linguistic structure
- for efficiency

Transformers

In each attention head:

$$\bar{V} = \textbf{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right)V.$$



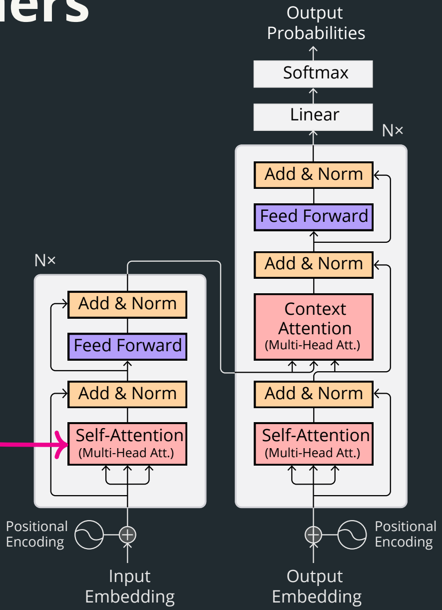
Transformers

In each attention head:

$$\bar{V} = \textbf{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V.$$

Attention in three places:

- Self-attention in the encoder



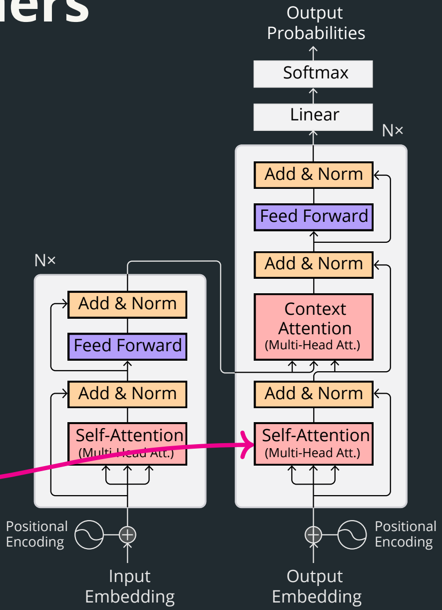
Transformers

In each attention head:

$$\bar{\mathbf{V}} = \mathbf{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}.$$

Attention in three places:

- Self-attention in the encoder
- Self-attention in the decoder



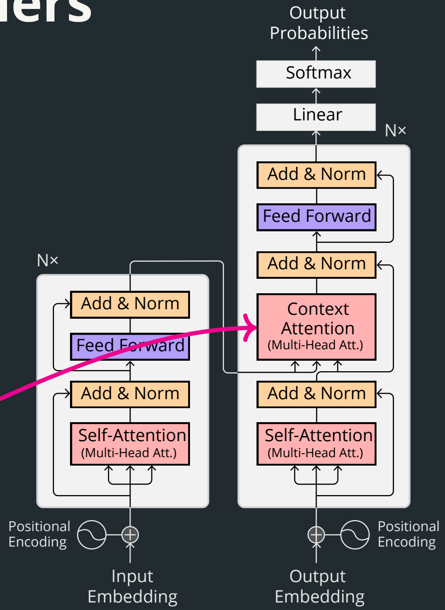
Transformers

In each attention head:

$$\bar{V} = \mathbf{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{d_k}}\right)\mathbf{V}.$$

Attention in three places:

- Self-attention in the encoder
- Self-attention in the decoder
- Contextual attention



Sparse Transformers

Sparse Transformers

Key idea: replace softmax in attention heads by a sparse normalizing function! 🙌

Adaptively Sparse Transformers

Key idea: replace softmax in attention heads by a sparse normalizing function! 🙌

Another key idea: use a normalizing function that is adaptively sparse via a learnable α! 🙌 🙌 🙌

What is softmax?

Softmax exponentiates and normalizes:

$$[\mathbf{softmax}(z)]_i := \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

What is softmax?

Softmax exponentiates and normalizes:

$$[\mathbf{softmax}(z)]_i := \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

It's fully dense: $\mathbf{softmax}(z) > \mathbf{0}$

(Peters et al., 2019)

α-entmax

Parametrized by $\alpha \geq 0$:

α-entmax

Parametrized by $\alpha \geq 0$:

- **Argmax** corresponds to $\alpha \to \infty$

α-entmax

Parametrized by $\alpha \geq 0$:

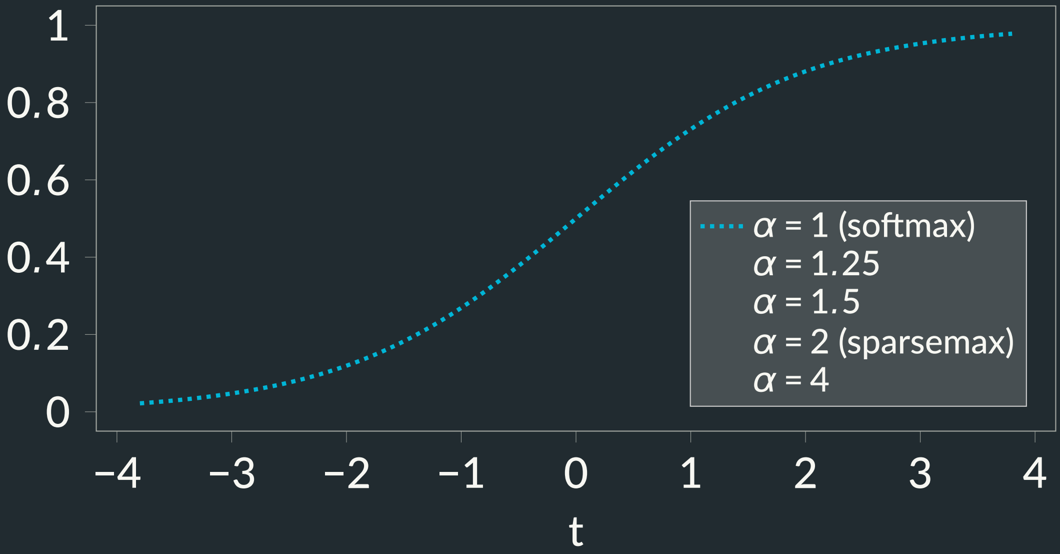
- **Argmax** corresponds to $\alpha \to \infty$
- **Softmax** amounts to $\alpha \to 1$

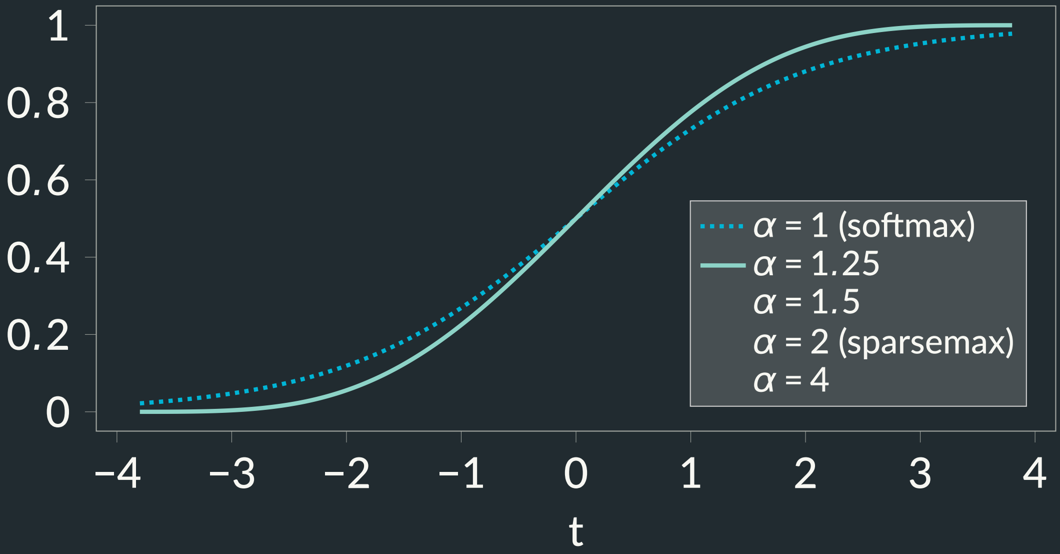
α-entmax

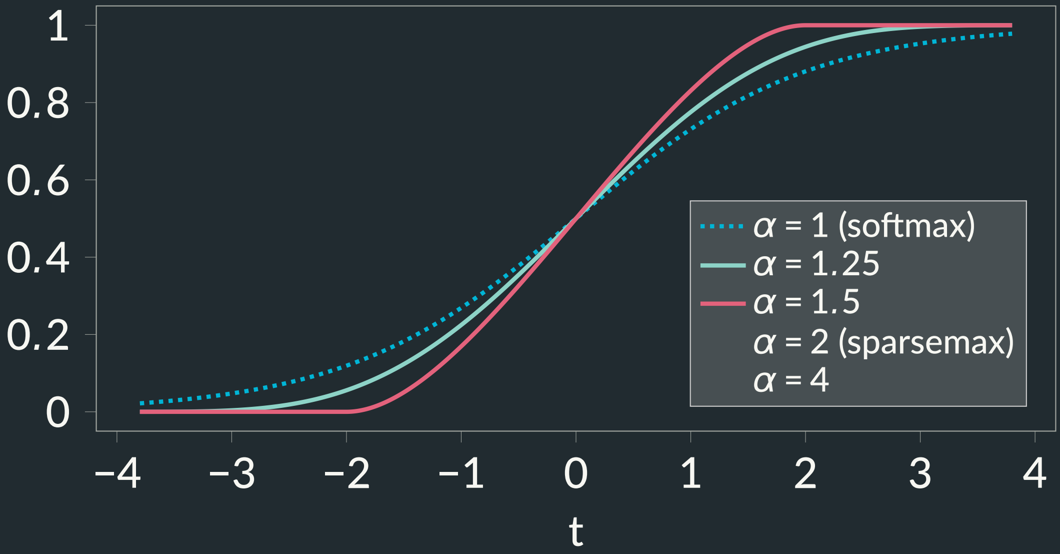
Parametrized by $\alpha \geq 0$:

- **Argmax** corresponds to $\alpha \to \infty$
- **Softmax** amounts to $\alpha \to 1$

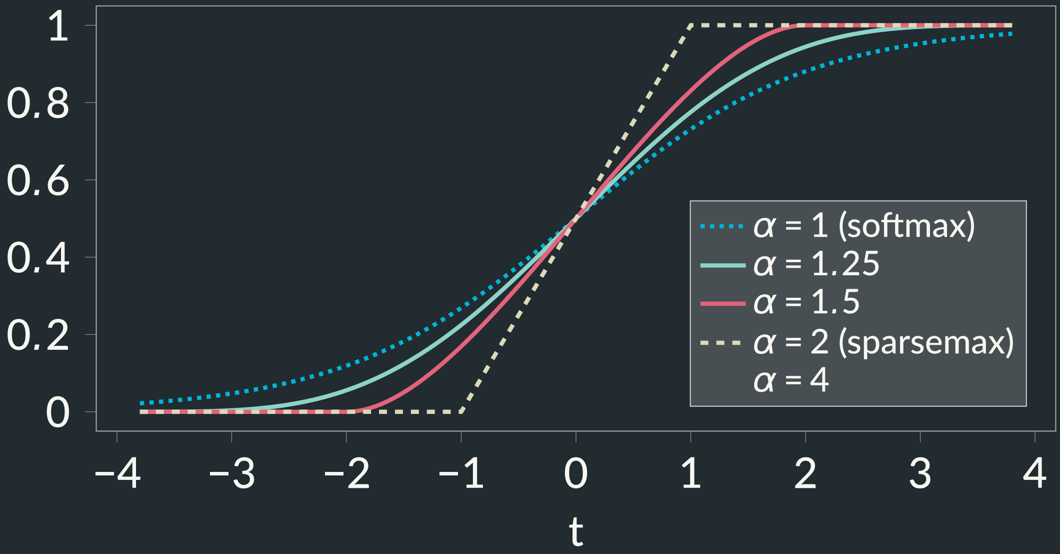
Key result: **can be sparse for** $\alpha > 1$, propensity for sparsity increases with α.

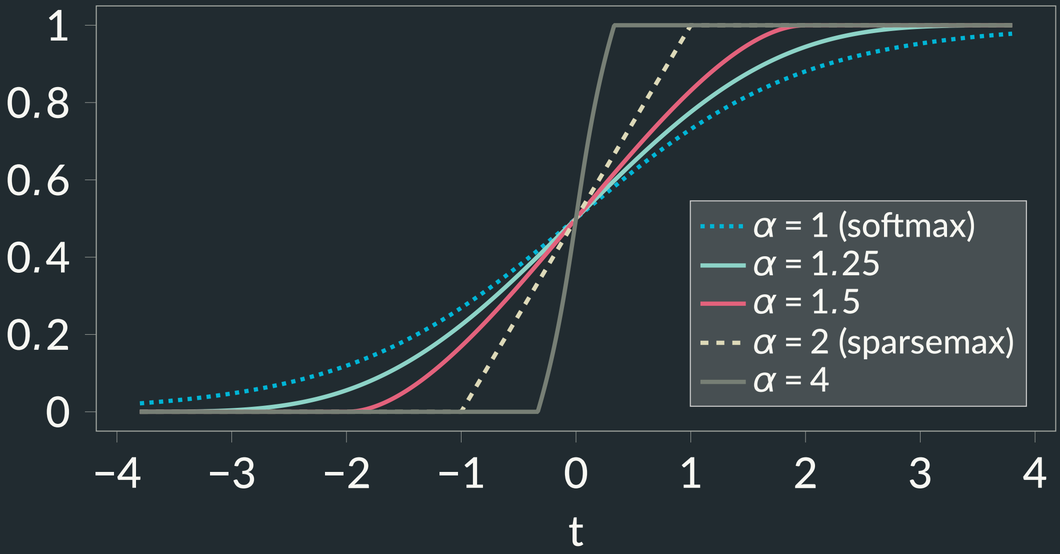

1
0.8
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4


1
0.8
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4



1
0.8
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4


1
0.8
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4


1
0.8
0.6
0.4
0.2
0
−4
−3
−2
−1
0
1
2
3
4
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4

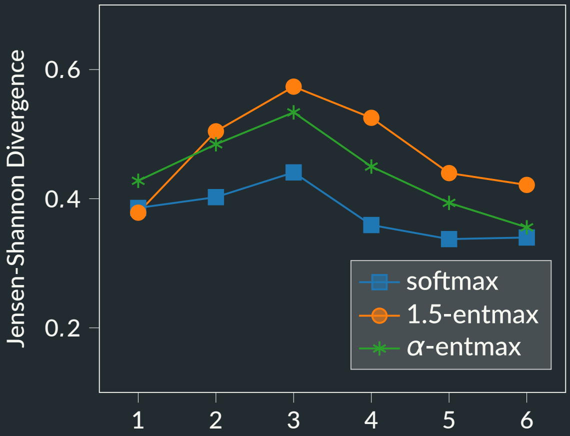
Learning α

Learning α

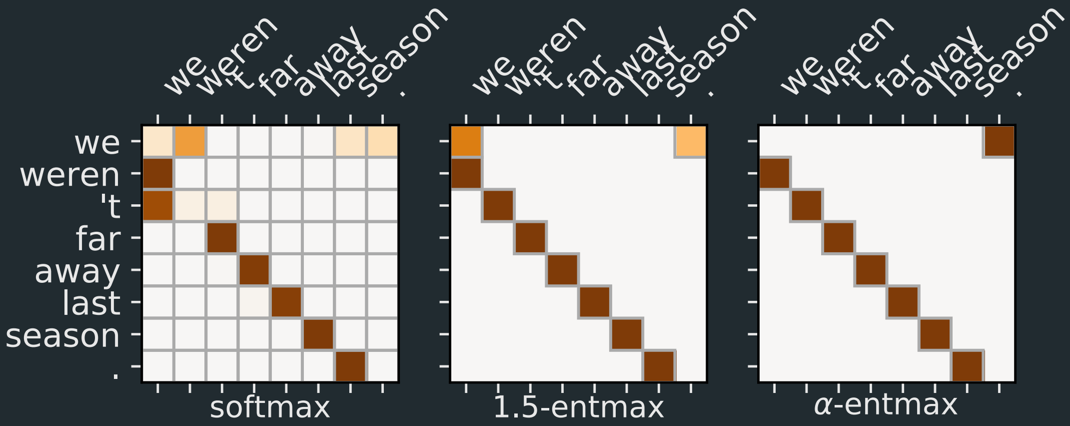
Key contribution:

a closed-form expression for $\dfrac{\partial\, \alpha\text{-entmax}(\mathbf{z})}{\partial \alpha}$ 😱

Head diversity per layer

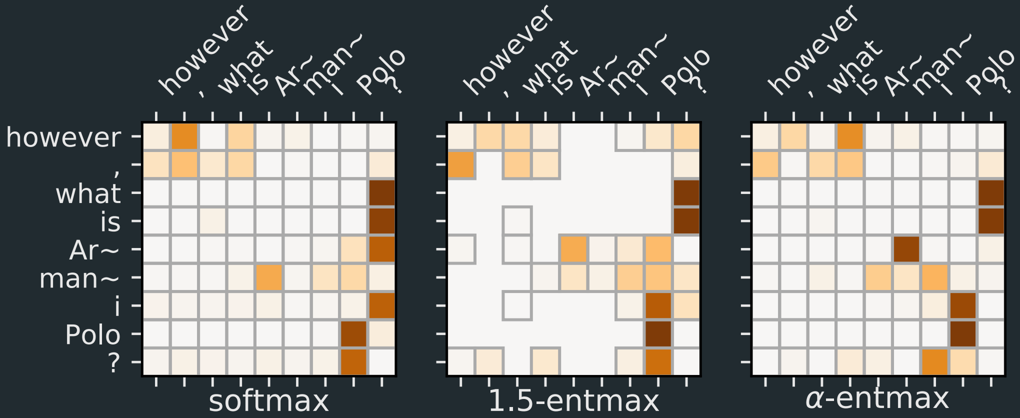


Previous position head



This head role was also found in Voita et al. (2019)! Learned $\alpha = 1.91$.

Interrogation-detecting head



Learned $\alpha = 1.05$.

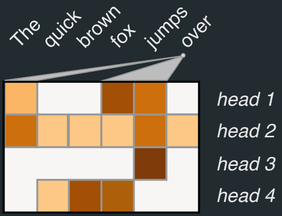
Conclusions and impact

Introduce adaptive sparsity
for Transformers via α-entmax with a gradient learnable α.

Conclusions and impact

Introduce adaptive sparsity
for Transformers via α-entmax with a gradient learnable α.

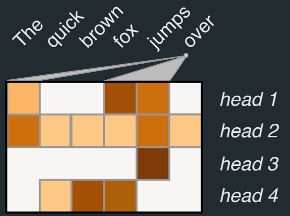
adaptive sparsity



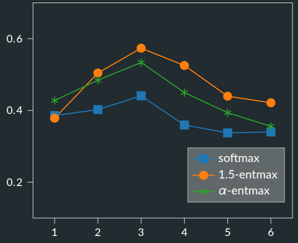
Conclusions and impact

Introduce adaptive sparsity
for Transformers via α-entmax with a gradient learnable α.

adaptive sparsity



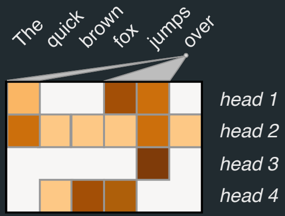
reduced head redundancy



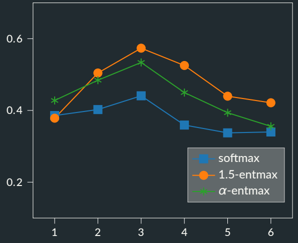
Conclusions and impact

Introduce adaptive sparsity
for Transformers via α-entmax with a gradient learnable α.

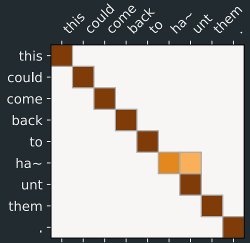
adaptive sparsity



reduced head redundancy



clearer head roles



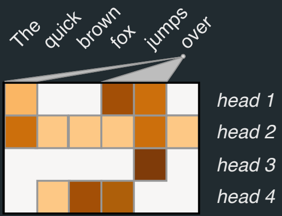
(Treviso et al., 2022; Zhang et al., 2021; Raganato et al., 2020)

Conclusions and impact

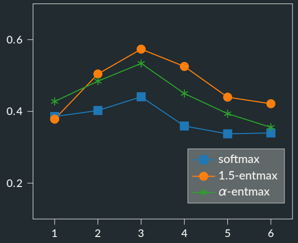
Subsequent work has:

- focused on taking computational advantage of sparsity
- proposed other sparse activations (*e.g.*, ReLU)
- incorporated fixed attention patterns that we found.

adaptive sparsity



reduced head redundancy



clearer head roles

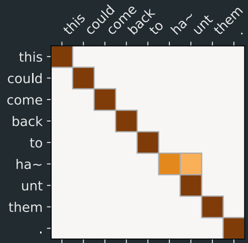


Table of Contents

Latent variable models

Latent variable z can be

Latent variable models

Latent variable z can be continuous

Source: Bouges et al., 2013

Latent variable models

Latent variable z can be continuous, discrete

Source: valleyeyecareaz.com

Latent variable models

Latent variable z can be continuous, discrete, or structured

Source: Liu et al., 2015

Training discrete or structured latent variable models

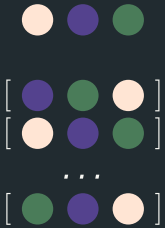
Latent variable z can be

Training discrete or structured latent variable models

Latent variable z can be discrete

Training discrete or structured latent variable models

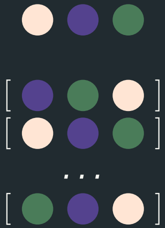
Latent variable z can be discrete or structured



Training discrete or structured latent variable models

Latent variable z can be discrete or structured

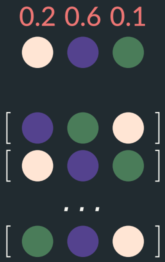
$\pi(z|x, \theta)$: distribution over possible z



Training discrete or structured latent variable models

Latent variable z can be discrete or structured

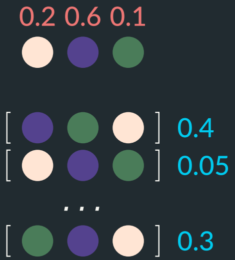
$\pi(z|x, \theta)$: distribution over possible z



Training discrete or structured latent variable models

Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z



Training discrete or structured latent variable models

Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z

$\ell(x, z; \theta)$: downstream loss: ELBO, Log-Likelihood, (...)

Training discrete or structured latent variable models

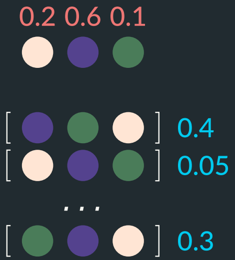
Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z

$\ell(x, z; \theta)$: downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:

Training discrete or structured latent variable models



Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z

$\ell(x, z; \theta)$: downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta)\, \ell(x, z; \theta)$$

Training discrete or structured latent variable models

Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z

$\ell(x, z; \theta)$: downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta)\, \ell(x, z; \theta)$$

If $\mathcal{Z}$ is large, this sum can get very expensive due to $\ell(x, z; \theta)$!

Training discrete or structured latent variable models

Latent variable z can be discrete or structured

$\pi(z|x, \theta)$: distribution over possible z

$\ell(x, z; \theta)$: downstream loss: ELBO, Log-Likelihood, (...)

To train, we need to compute the following expectation:

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x, \theta)\, \ell(x, z; \theta)$$

If $\mathcal{Z}$ is combinatorial, this can be intractable to compute! 😱 😱 😱

Current solutions

Using emergent communication as example

Method	success (%)	Dec. calls
Monte Carlo		
Marginalization		
Dense	93.37 ±0.42	256

Current solutions

Using emergent communication as example

- SFE → unbiased but high variance

Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
Marginalization		
Dense	93.37 ±0.42	256

Current solutions

Using emergent communication as example

- SFE → unbiased but high variance

Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Marginalization		
Dense	93.37 ±0.42	256

Current solutions

Using emergent communication as example

- SFE → unbiased but high variance
- Gumbel-Softmax → continuous relaxation, biased estimation

Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Gumbel	23.51 ±16.19	1
ST Gumbel	27.42 ±13.36	1
Marginalization		
Dense	93.37 ±0.42	256

Current solutions

Using emergent communication as example

- SFE → unbiased but high variance
- Gumbel-Softmax → continuous relaxation, biased estimation

New option: use sparsity! 🙌

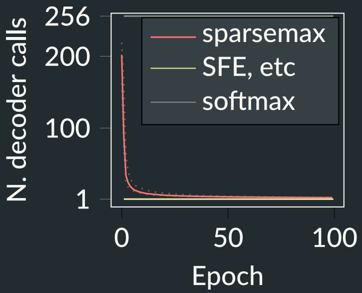
Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Gumbel	23.51 ±16.19	1
ST Gumbel	27.42 ±13.36	1
Marginalization		
Dense	93.37 ±0.42	256
Sparse	93.35 ±0.50	3.13±0.48

Current solutions

Using emergent communication as example

- SFE → unbiased but high variance
- Gumbel-Softmax → continuous relaxation, biased estimation

New option: use sparsity! 🙌



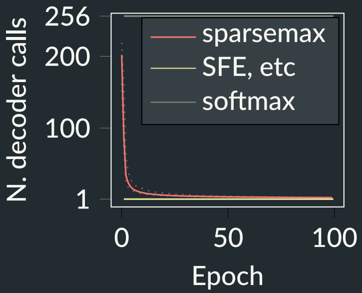
Current solutions

Using emergent communication as example

- SFE → unbiased but high variance
- Gumbel-Softmax → continuous relaxation, biased estimation

New option: use sparsity! 🙌

no need for sampling → no variance



Current solutions

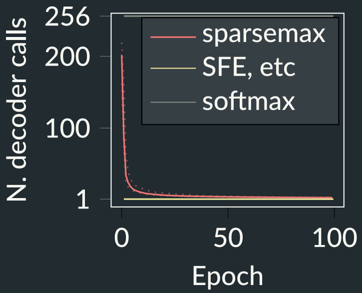
Using emergent communication as example

- SFE → unbiased but high variance
- Gumbel-Softmax → continuous relaxation, biased estimation

New option: use sparsity! 🙌

no need for sampling → no variance

no continuous relaxation



Taking a step back...

Does the expectation over possible z need to be expensive?

Taking a step back...

Does the expectation over possible z need to be expensive?

$$
\begin{aligned}
\mathcal{L}_x(\theta) &= \sum_{z \in \mathcal{Z}} \pi(z|x, \theta)\, \ell(x, z; \theta) \\
&= \pi(z_1|x, \theta)\, \ell(x, z_1; \theta) + \pi(z_2|x, \theta)\, \ell(x, z_2; \theta) + \ldots \\
&+ \pi(z_i|x, \theta)\, \ell(x, z_i; \theta) + \ldots + \pi(z_N|x, \theta)\, \ell(x, z_N; \theta)
\end{aligned}
$$

Taking a step back...

Does the expectation over possible z need to be expensive?

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z\in\mathcal{Z}} \pi(z|x,\theta)\,\ell(x,z;\theta)\\ &= \pi(z_1|x,\theta)\,\ell(x,z_1;\theta) + \pi(z_2|x,\theta)\,\ell(x,z_2;\theta) + \ldots\\ &+ \pi(z_i|x,\theta)\,\ell(x,z_i;\theta) + \ldots + \pi(z_N|x,\theta)\,\ell(x,z_N;\theta)\end{aligned}$$

Usually we normalize π with softmax $\propto \exp(s) \Rightarrow \pi(z_i|x,\theta) > 0$

(Martins and Astudillo, 2016; Niculae, Martins, et al., 2018)

Sparse normalizers

We use sparsemax, top-k sparsemax and SparseMAP to allow efficient marginalization

(Martins and Astudillo, 2016; Niculae, Martins, et al., 2018)

Sparse normalizers

We use sparsemax, top-k sparsemax and SparseMAP to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero**!

(Martins and Astudillo, 2016; Niculae, Martins, et al., 2018)

Sparse normalizers

We use sparsemax, top-k sparsemax and SparseMAP to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero**!

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x,\theta)\,\ell(x,z;\theta)$$
$$= \pi(z_1|x,\theta)\,\ell(x,z_1;\theta) + \underbrace{\pi(z_2|x,\theta)}_{=0}\,\ell(x,z_2;\theta) + \ldots$$
$$+ \pi(z_i|x,\theta)\,\ell(x,z_i;\theta) + \ldots + \underbrace{\pi(z_N|x,\theta)}_{=0}\,\ell(x,z_N;\theta)$$

Sparse normalizers

We use sparsemax, top-*k* sparsemax and SparseMAP to allow efficient marginalization

These functions are able to assign **probabilities of exactly zero**!

$$\begin{aligned}\mathcal{L}_x(\theta) &= \sum_{z\in\mathcal{Z}} \pi(z|x,\theta)\,\ell(x,z;\theta)\\ &= \pi(z_1|x,\theta)\,\ell(x,z_1;\theta) + \underbrace{\pi(z_2|x,\theta)}_{=0}\,\cancel{\ell(x,z_2;\theta)} + \ldots\\ &+ \pi(z_i|x,\theta)\,\ell(x,z_i;\theta) + \ldots + \underbrace{\pi(z_N|x,\theta)}_{=0}\,\cancel{\ell(x,z_N;\theta)}\end{aligned}$$

No need for computing $\ell(x,z;\theta)$ for all $z\in\mathcal{Z}$!

Results

We test our methods for models with discrete latent variables,

Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE

Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of $\mathcal{Z}$,

Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of $\mathcal{Z}$,

- Bit-vector VAE

Results

We test our methods for models with discrete latent variables,

- Semi-Supervised VAE
- Emergent communication

but also in models with an exponentially large set of $\mathcal{Z}$,

- Bit-vector VAE

Our methods are top-performers and efficient!

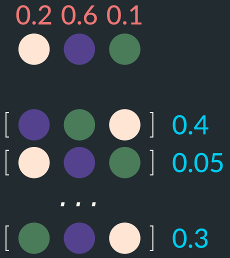
Key takeaways

We introduce a new method
to train latent variable models.

Key takeaways

We introduce a new method
to train latent variable models.

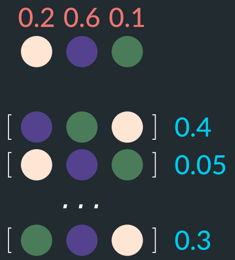
discrete and structured



Key takeaways

We introduce a new method
to train latent variable models.

discrete and structured



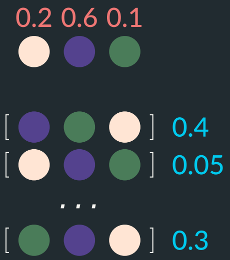
deterministic, yet efficient

$$
\begin{aligned}
\mathcal{L}_x(\theta) = \; & \pi(z_1|x,\theta)\,\ell(x,z_1;\theta) \\
& + \underbrace{\pi(z_2|x,\theta)}_{=0}\,\cancel{\ell(x,z_2;\theta)} \\
& + \dots + \pi(z_i|x,\theta)\,\ell(x,z_i;\theta) \\
& + \dots + \underbrace{\pi(z_N|x,\theta)}_{=0}\,\cancel{\ell(x,z_N;\theta)}
\end{aligned}
$$

Key takeaways

We introduce a new method
to train latent variable models.

discrete and structured



deterministic, yet efficient

$$\begin{aligned}\mathcal{L}_x(\theta) = {} & \pi(z_1|x,\theta)\;\ell(x,z_1;\theta)\\ & + \underbrace{\pi(z_2|x,\theta)}_{=0}\;\cancel{\ell(x,z_2;\theta)}\\ & + \dots + \pi(z_i|x,\theta)\;\ell(x,z_i;\theta)\\ & + \dots + \underbrace{\pi(z_N|x,\theta)}_{=0}\;\cancel{\ell(x,z_N;\theta)}\end{aligned}$$

sparse, as needed

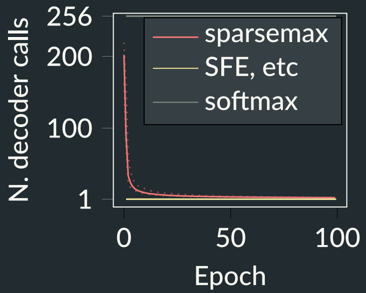


Table of Contents

Conclusions

Using learned sparsity and weak supervision
we took steps to take neural models closer to version 2.0

Conclusions

Using learned sparsity and weak supervision
we took steps to take neural models closer to version 2.0

data-efficiency

model (data size)	BLEU↑
dual-source transformer (8M)	71.72
dual-source transformer (23K)	59.78
ours (23K)	70.66

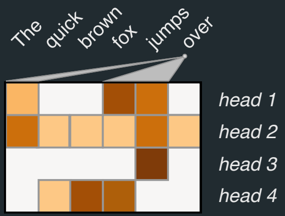
Conclusions

Using learned sparsity and weak supervision
we took steps to take neural models closer to version 2.0

data-efficiency

model (data size)	BLEU↑
dual-source transformer (8M)	71.72
dual-source transformer (23K)	59.78
ours (23K)	70.66

transparency



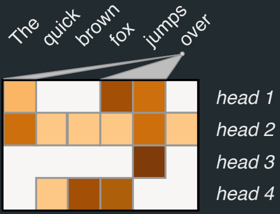
Conclusions

Using learned sparsity and weak supervision
we took steps to take neural models closer to version 2.0

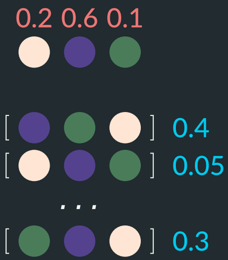
data-efficiency

model (data size)	BLEU↑
dual-source transformer (8M)	71.72
dual-source transformer (23K)	59.78
ours (23K)	70.66

transparency



better compactness



References I

- Bouges, Pierre, Thierry Chateau, Christophe Blanc, and Gaëlle Loosli (Dec. 2013). "Handling missing weak classifiers in boosted cascade: application to multiview and occluded face detection". In: *EURASIP Journal on Image and Video Processing* 2013, p. 55. DOI: 10.1186/1687-5281-2013-55.
- Correia, Gonçalo M, Vlad Niculae, and André FT Martins (2019). "Adaptively sparse transformers". In: *Proc. EMNLP*.
- Correia, Gonçalo M. and André F. T. Martins (July 2019). "A Simple and Effective Approach to Automatic Post-Editing with Transfer Learning". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3050–3056. DOI: 10.18653/v1/P19-1292. URL: https://www.aclweb.org/anthology/P19-1292.
- Correia, Gonçalo M., Vlad Niculae, Wilker Aziz, and André F. T. Martins (2020). "Efficient Marginalization of Discrete and Structured Latent Variables via Sparsity". In: *Proc. NeurIPS*. URL: https://arxiv.org/abs/2007.01919.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *Proc. NAACL-HLT*.
- Junczys-Dowmunt, Marcin and Roman Grundkiewicz (2018). "MS-UEdin Submission to the WMT2018 APE Shared Task: Dual-Source Transformer for Automatic Post-Editing". In: *Proceedings of WMT18*.
- Kodama, Takashi, Ryuichiro Higashinaka, Koh Mitsuda, Ryo Masumura, Yushi Aono, Ryuta Nakamura, Noritake Adachi, and Hidetoshi Kawabata (2020). "Generating Responses That Reflect Meta Information in User-Generated Question Answer Pairs". In: *Proceedings of LREC*.

References II

Lazaridou, Angeliki, Alexander Peysakhovich, and Marco Baroni (2017). "Multi-agent cooperation and the emergence of (natural) language". In: *Proc. ICLR*.

Lee, Jihyung, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee (2020). "POSTECH-ETRI's Submission to the WMT2020 APE Shared Task: Automatic Post-Editing with Cross-lingual Language Model". In: *Proceedings of WMT*.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (Dec. 2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Martins, André FT and Ramón Fernandez Astudillo (2016). "From softmax to sparsemax: A sparse model of attention and multi-label classification". In: *Proc. of ICML*.

Niculae, Vlad and Mathieu Blondel (2017). "A Regularized Framework for Sparse and Structured Neural Attention". In: *arXiv preprint arXiv:1705.07704*.

Niculae, Vlad, André FT Martins, Mathieu Blondel, and Claire Cardie (2018). "SparseMAP: Differentiable sparse structured inference". In: *Proc. of ICML*.

Peters, Ben, Vlad Niculae, and André F. T. Martins (2019). "Sparse Sequence-to-Sequence Models". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Raganato, Alessandro, Yves Scherrer, and Jörg Tiedemann (2020). "Fixed Encoder Self-Attention Patterns in Transformer-Based Machine Translation". In: *Proceedings of EMNLP*.

References III

- Treviso, Marcos, António Góis, Patrick Fernandes, Erick Fonseca, and André F. T. Martins (2022). "Predicting Attention Sparsity in Transformers". In: *Proceedings of SPNLP*.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need". In: *Proc. of NeurIPS*.
- Voita, Elena, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov (2019). "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned". In: *Proc. ACL*.
- Zhang, Biao, Ivan Titov, and Rico Sennrich (2021). "Sparse Attention with Linear Units". In: *Proceedings of EMNLP*.

Parameter sharing analysis

	TER↓	BLEU↑
MT Baseline	24.76	62.11
Transformer	27.80	60.76
Transformer decoder	20.33	69.31
Pre-trained BERT	20.83	69.11
with CA ← SA	18.91	71.81
and SA ⟷ Encoder SA	**18.44**	**72.25**
and CA ⟷ SA	18.75	71.83
and FF ⟷ Encoder FF	19.04	71.53

(Niculae and Blondel, 2017)

Ω-Regularized Argmax

For convex Ω, define the **Ω-regularized argmax transformation:**

$$\mathbf{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{z}^{\top}\boldsymbol{p} - \Omega(\boldsymbol{p})$$

(Niculae and Blondel, 2017)

Ω-Regularized Argmax

For convex Ω, define the **Ω-regularized argmax transformation:**

$$\textbf{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\Delta} \boldsymbol{z}^{\top}\boldsymbol{p} - \Omega(\boldsymbol{p})$$

- Argmax corresponds to **no regularization,** $\Omega \equiv 0$

(Niculae and Blondel, 2017)

Ω-Regularized Argmax

For convex Ω, define the **Ω-regularized argmax transformation:**

$$\mathbf{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p} \in \Delta} \boldsymbol{z}^{\top}\boldsymbol{p} - \Omega(\boldsymbol{p})$$

- Argmax corresponds to **no regularization**, $\Omega \equiv 0$
- Softmax amounts to **entropic regularization**, $\Omega(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$

(Niculae and Blondel, 2017)

Ω-Regularized Argmax

For convex Ω, define the **Ω-regularized argmax transformation:**

$$\mathbf{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\Delta} \boldsymbol{z}^{\top}\boldsymbol{p} - \Omega(\boldsymbol{p})$$

- Argmax corresponds to **no regularization,** $\Omega \equiv 0$
- Softmax amounts to **entropic regularization,** $\Omega(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$
- Sparsemax amounts to **ℓ_2-regularization,** $\Omega(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}\|^2$.

(Martins and Astudillo, 2016)

Ω-Regularized Argmax

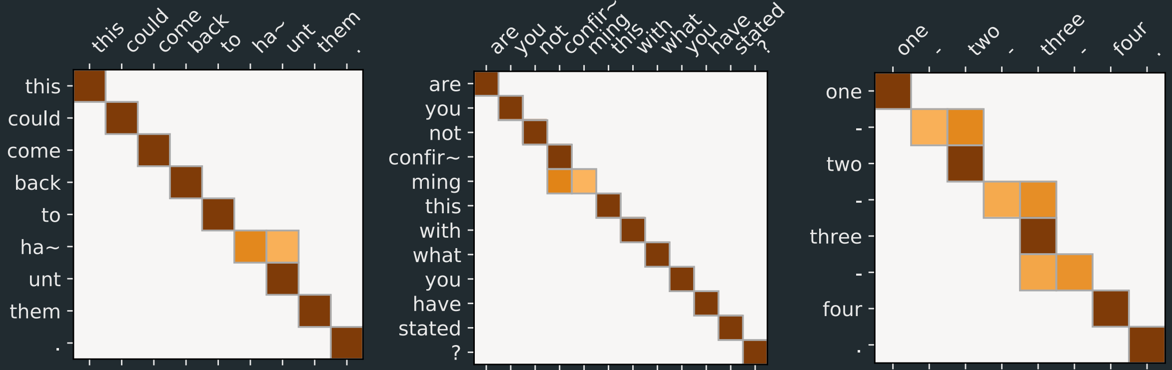
For convex Ω, define the Ω**-regularized argmax transformation:**

$$\mathbf{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\Delta} \boldsymbol{z}^\top \boldsymbol{p} - \Omega(\boldsymbol{p})$$

- Argmax corresponds to **no regularization,** $\Omega \equiv 0$
- Softmax amounts to **entropic regularization,** $\Omega(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$
- Sparsemax amounts to ℓ_2**-regularization,** $\Omega(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}\|^2$.

Is there something in-between?

Subword-Merging Head



Learned $\alpha = 1.91$.

Semi-Supervised VAE

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$
$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- Semi-Supervised VAE on MNIST: z is one of 10 categories

Semi-Supervised VAE

Gaussian VAE

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

classification network

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- Semi-Supervised VAE on MNIST: z is one of 10 categories

Semi-Supervised VAE

sum over the 10 digits

Gaussian VAE

classification network

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- Semi-Supervised VAE on MNIST: z is one of 10 categories

Semi-Supervised VAE

Gaussian VAE

sum over
the 10 digits

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

classification network

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- Semi-Supervised VAE on MNIST: z is one of 10 categories
- Train this with 10% labeled data

Semi-Supervised VAE

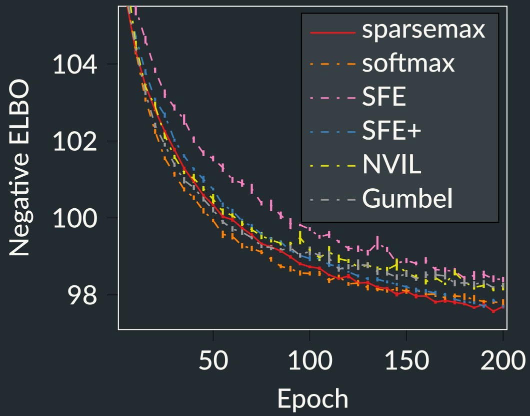
Method	Accuracy (%)	Dec. calls
Monte Carlo		
SFE	$94.75_{\pm .002}$	1
SFE+	$96.53_{\pm .001}$	2
NVIL	$96.01_{\pm .002}$	1
Gumbel	$95.46_{\pm .001}$	1
Marginalization		
Dense	$96.93_{\pm .001}$	10

Semi-Supervised VAE

Method	Accuracy (%)	Dec. calls
Monte Carlo		
SFE	94.75 $\pm$.002	1
SFE+	96.53 $\pm$.001	2
NVIL	96.01 $\pm$.002	1
Gumbel	95.46 $\pm$.001	1
Marginalization		
Dense	96.93 $\pm$.001	10
Sparse	96.87 $\pm$.001	1.01 $\pm$0.01

Semi-Supervised VAE

Method	Accuracy (%)	Dec. calls
Monte Carlo		
SFE	$94.75_{\pm.002}$	1
SFE+	$96.53_{\pm.001}$	2
NVIL	$96.01_{\pm.002}$	1
Gumbel	$95.46_{\pm.001}$	1
Marginalization		
Dense	$96.93_{\pm.001}$	10
Sparse	$96.87_{\pm.001}$	$1.01_{\pm0.01}$



Emergent communication

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$
$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

Emergent communication

sender

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

receiver

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- receiver picks image from a set $\mathcal{V}$ based on message

Emergent communication

sum over
all possible messages
in the vocabulary

sender

receiver

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- receiver picks image from a set $\mathcal{V}$ based on message
- images come from ImageNet

(Lazaridou et al., 2017; Correia, Niculae, Aziz, et al., 2020)

Emergent Communication

... but make it harder: $|\mathcal{Z}| = 256$, $|\mathcal{V}| = 16$

Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Gumbel	23.51 ±16.19	1
ST Gumbel	27.42 ±13.36	1
Marginalization		

(Lazaridou et al., 2017; Correia, Niculae, Aziz, et al., 2020)

Emergent Communication

... but make it harder: $|\mathcal{Z}| = 256$, $|\mathcal{V}| = 16$

Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Gumbel	23.51 ±16.19	1
ST Gumbel	27.42 ±13.36	1
Marginalization		
Dense	93.37 ±0.42	256

(Lazaridou et al., 2017; Correia, Niculae, Aziz, et al., 2020)

Emergent Communication

... but make it harder: $|\mathcal{Z}| = 256$, $|\mathcal{V}| = 16$

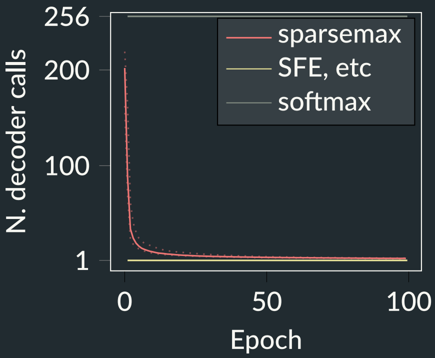
Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ± 2.84	1
SFE+	44.32 ± 2.72	2
NVIL	37.04 ± 1.61	1
Gumbel	23.51 ± 16.19	1
ST Gumbel	27.42 ± 13.36	1
Marginalization		
Dense	93.37 ± 0.42	256
Sparse	93.35 ± 0.50	3.13 ± 0.48

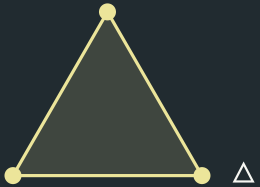
(Lazaridou et al., 2017; Correia, Niculae, Aziz, et al., 2020)

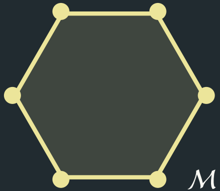
Emergent Communication

... but make it harder: $|\mathcal{Z}| = 256$, $|\mathcal{V}| = 16$

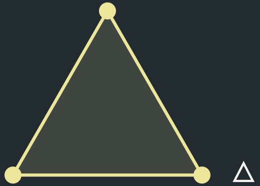
Method	success (%)	Dec. calls
Monte Carlo		
SFE	33.05 ±2.84	1
SFE+	44.32 ±2.72	2
NVIL	37.04 ±1.61	1
Gumbel	23.51 ±16.19	1
ST Gumbel	27.42 ±13.36	1
Marginalization		
Dense	93.37 ±0.42	256
Sparse	93.35 ±0.50	3.13±0.48

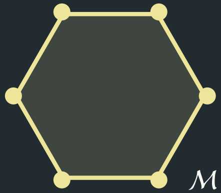



△

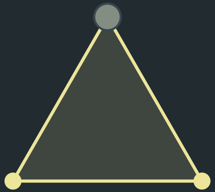

$\mathcal{M}$

$$\begin{aligned}\mathcal{M} &:= \operatorname{conv}\left\{\boldsymbol{a}_z : z \in \mathcal{Z}\right\} \\ &= \left\{\boldsymbol{A}\boldsymbol{p} : \boldsymbol{p} \in \triangle\right\} \\ &= \left\{\mathbb{E}_{Z\sim\boldsymbol{p}}\, \boldsymbol{a}_Z : \boldsymbol{p} \in \triangle\right\}\end{aligned}$$

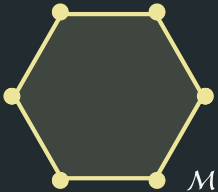




- **argmax** $\displaystyle \arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{s}$

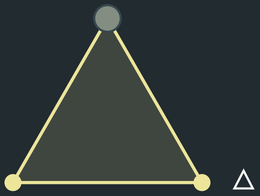


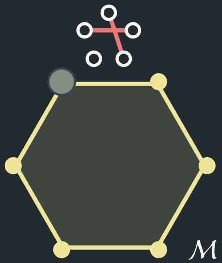
$\triangle$



$\mathcal{M}$

- **argmax** $\underset{\boldsymbol{p}\in\triangle}{\arg\max}\, \boldsymbol{p}^\top \boldsymbol{s}$
- **MAP** $\underset{\boldsymbol{\mu}\in\mathcal{M}}{\arg\max}\, \boldsymbol{\mu}^\top \boldsymbol{t}$

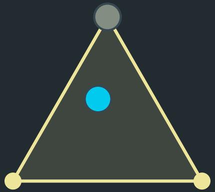


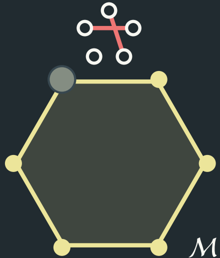


argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s}$

softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} + \mathrm{H}(\boldsymbol{p})$

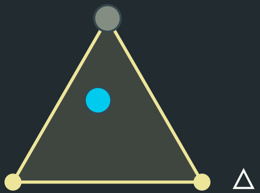
MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t}$

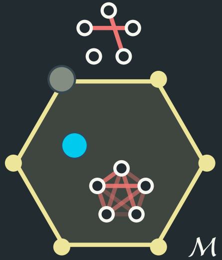




- **argmax** $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s}$
- **softmax** $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} + \mathrm{H}(\boldsymbol{p})$

- **MAP** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t}$
- **marginals** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$





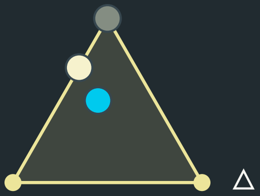
argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s}$

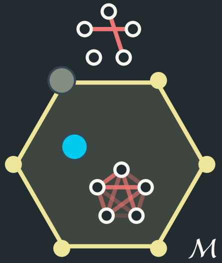
softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} + \mathrm{H}(\boldsymbol{p})$

sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} - 1/2\|\boldsymbol{p}\|^2$

MAP $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t}$

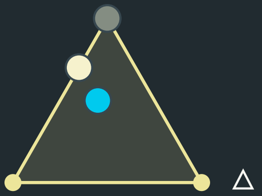
marginals $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

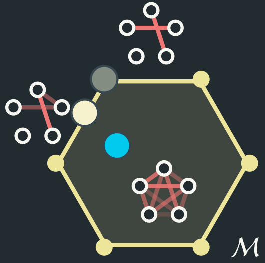




(Niculae, Martins, Blondel, and Cardie, 2018)

argmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s}$	**MAP** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t}$
softmax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} + \mathrm{H}(\boldsymbol{p})$	**marginals** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t} + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$
sparsemax $\arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{p}^\top \boldsymbol{s} - 1/2\lVert\boldsymbol{p}\rVert^2$	**SparseMAP** $\arg\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{t} - 1/2\lVert\boldsymbol{\mu}\rVert^2$





Bit-vector VAE

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$
$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

Bit-vector VAE

generative network

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

inference network

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- VAE where z is a collection of D bits

Bit-vector VAE

generative network

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$

inference network

$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

- VAE where z is a collection of D bits
- Minimize the negative ELBO

Bit-vector VAE

sum over
an exponetially large
set of structures

generative network

inference network

$$\mathcal{L}_x(\theta) = \sum_{z \in \mathcal{Z}} \pi(z|x)\, \ell(x, z)$$
$$= \mathbb{E}_{z \sim \pi(z|x)}\, \ell(x, z)$$

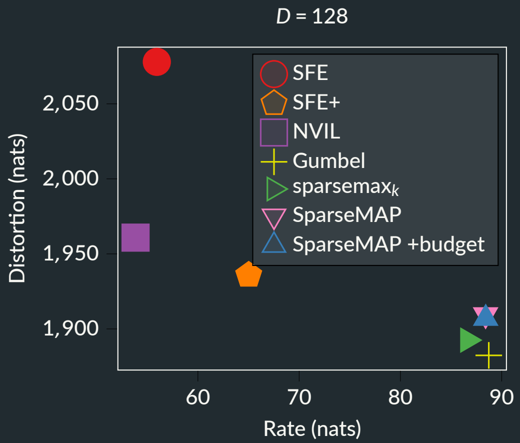
- VAE where z is a collection of D bits
- Minimize the negative ELBO

Bit-vector VAE

Method	$D = 32$	$D = 128$
Monte Carlo		
SFE	3.74	3.77
SFE+	3.61	3.59
NVIL	3.65	3.60
Gumbel	3.57	3.49
Marginalization		
Top-*k* sparsemax	3.62	3.61
SparseMAP	3.72	3.67
SparseMAP (w/ budget)	3.64	3.66

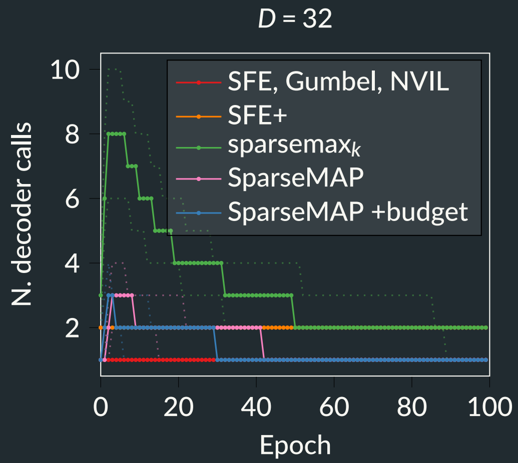
Bit-vector VAE

Method	$D = 32$	$D = 128$
Monte Carlo		
SFE	3.74	3.77
SFE+	3.61	3.59
NVIL	3.65	3.60
Gumbel	3.57	3.49
Marginalization		
Top-k sparsemax	3.62	3.61
SparseMAP	3.72	3.67
SparseMAP (w/ budget)	3.64	3.66



Bit-vector VAE

Method	$D = 32$	$D = 128$
Monte Carlo		
SFE	3.74	3.77
SFE+	3.61	3.59
NVIL	3.65	3.60
Gumbel	3.57	3.49
Marginalization		
Top-k sparsemax	3.62	3.61
SparseMAP	3.72	3.67
SparseMAP (w/ budget)	3.64	3.66



Bit-vector VAE

Method	$D = 32$	$D = 128$
Monte Carlo		
SFE	3.74	3.77
SFE+	3.61	3.59
NVIL	3.65	3.60
Gumbel	3.57	3.49
Marginalization		
Top-*k* sparsemax	3.62	3.61
SparseMAP	3.72	3.67
SparseMAP (w/ budget)	3.64	3.66