Masked Pre-training through the Bayesian lens

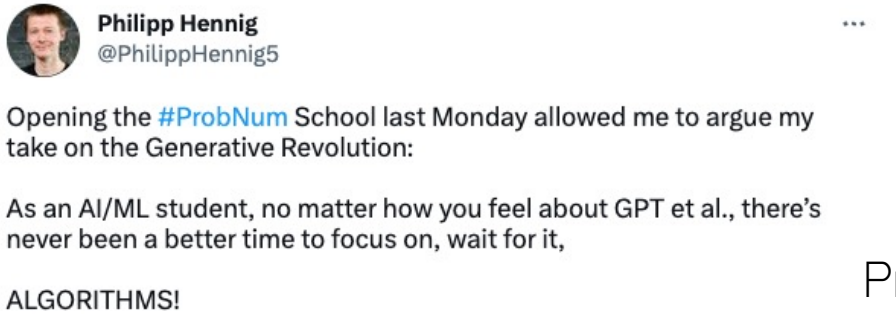
Pablo Moreno-Muñoz
pabmo@dtu.dk

Surfing the GPT tsunami

PhD students

Surfing the GPT tsunami

PhD students



Profs and PIs

Surfing the GPT tsunami

PhD students

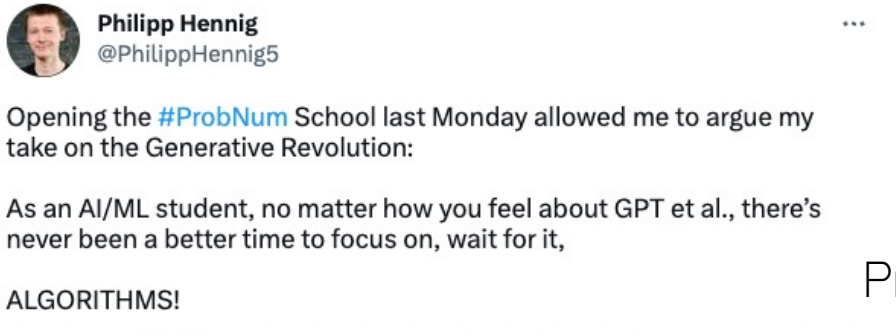
Reality

Profs and PIs

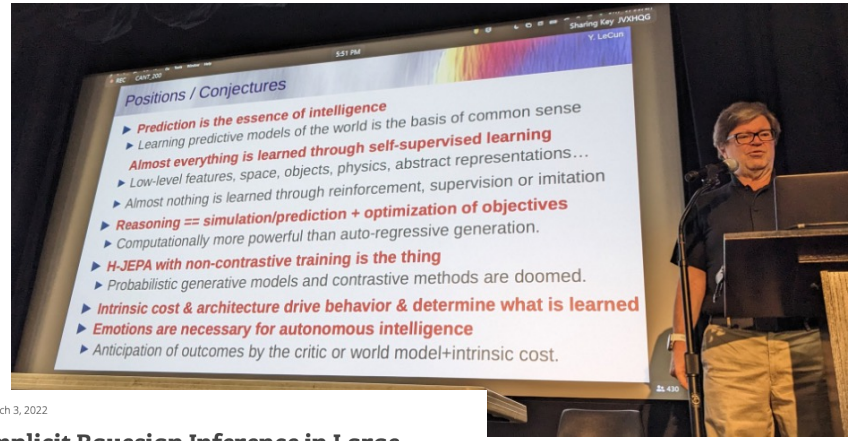
Surfing the GPT tsunami

PhD students

Reality



Profs and PIs



inFERENCe

March 3, 2022

Implicit Bayesian Inference in Large Language Models

This intriguing paper kept me thinking long enough for me to I decide it's time to resurrect my blogging (I started writing this during ICLR review period, and realised it might be a good idea to wait until that's concluded)

New insights

A new way of learning?

A new way of learning?

Yann LeCun
@ylecun

1. Self-Supervised Learning
2. ResNets (not intellectually deep, but useful)
3. Gating -> Attention -> Dynamic connection graphs.
4. Differentiable memory.
5. Permutation-equivariant modules, e.g. multihead self-attention -> Transformers.

Traducir Tweet

David Chalmers @davidchalmers42 · 13 sept. 2022
what are the most important intellectual breakthroughs (new ideas) in AI in the last ten years?
Mostrar este hilo

5:14 p. m. · 15 sept. 2022

A new way of learning?

Self-supervised Learning 101

"Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. The general technique of self-supervised learning is to predict any unobserved or hidden part (...) of the input from any observed or unhidden part of the input"

Meta AI

A new way of learning?

Self-supervised Learning 101

"Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. The general technique of self-supervised learning is to predict any unobserved or hidden part (...) of the input from any observed or unhidden part of the input"

Meta AI

This sentence brings us straight to Masked Pre-Training and its success in NLP

Example

8-dimensional observation

$$\boldsymbol{x}_i = \{ \text{The MLLS center conducts basic machine learning research} \}$$

each dimension is denoted as *token*

A new way of learning?

Self-supervised Learning 101

"Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. The general technique of self-supervised learning is to predict any unobserved or hidden part (...) of the input from any observed or unhidden part of the input"

Meta AI

This sentence brings us straight to Masked Pre-Training and its success in NLP

Example

8-dimensional observation

$\boldsymbol{x}_i = \{$ The [XXX] center conducts [XXX] [XXX] learning research $\}$ iteration 1

[XXX] denotes masking, and we are interested in maximising the sucess of predicting all [XXX]

A new way of learning?

Self-supervised Learning 101

"Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. The general technique of self-supervised learning is to predict any unobserved or hidden part (...) of the input from any observed or unhidden part of the input"

Meta AI

This sentence brings us straight to Masked Pre-Training and its success in NLP

Example

8-dimensional observation

$\boldsymbol{x}_i = \{$ The MLLS [XXX] [XXX] basic machine learning [XXX] $\}$ iteration 2

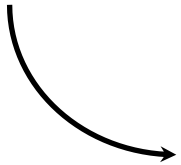
[XXX] denotes masking, and we are interested in maximising the sucess of predicting all [XXX]

A new way of learning?

Self-supervised Learning 101

"Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. The general technique of self-supervised learning is to predict any unobserved or hidden part (...) of the input from any observed or unhidden part of the input"

Meta AI



This sentence brings us straight to Masked Pre-Training and its success in NLP

Example

8-dimensional observation

$\boldsymbol{x}_i = \{$ The MLLS center conducts basic [XXX] [XXX] [XXX] $\}$ iteration 3

[XXX] denotes masking, and we are interested in maximising the sucess of predicting all [XXX]

Masked Pre-Training (MPT)

A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{The, center, conducts, learning, research})$$

Masked Pre-Training (MPT)

A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

$$\log p(\text{[XXX]}, \text{[XXX]}, \text{[XXX]} \mid \text{The}, \text{center}, \text{conducts}, \text{learning}, \text{research})$$

MLLS basic machine

Masked Pre-Training (MPT)

A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{The, center, conducts, learning, research}) \quad \text{iteration 1}$$

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{MLSS, center, basic, machine, learning}) \quad \text{iteration 2}$$

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{The, MLSS, center, conducts, basic}) \quad \text{iteration 3}$$

$$\vdots \qquad \vdots$$

Masked Pre-Training (MPT)

A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{The, center, conducts, learning, research}) \quad \text{iteration 1}$$

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{MLSS, center, basic, machine, learning}) \quad \text{iteration 2}$$

$$\log p(\text{[XXX], [XXX], [XXX]} \mid \text{The, MLSS, center, conducts, basic}) \quad \text{iteration 3}$$

$$\vdots \qquad \vdots$$

We are indeed maximising: $$\log p_\theta(\boldsymbol{x}_\mathcal{M} | \boldsymbol{x}_\mathcal{R}) = \sum_{t=1}^{M} \log p_\theta(x_{\mathcal{M}(t)} | \boldsymbol{x}_\mathcal{R})$$

Masked Pre-Training (MPT)

A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

2) A relatively large number of tools are used to set up a masked input, which in the end outputs the predicted tokens. For instance in BERT, these ones include an embedding of the NLP tokens into a real space, an encoder to a latent representation and a multi-head transformer, among others.

Masked Pre-Training (MPT)

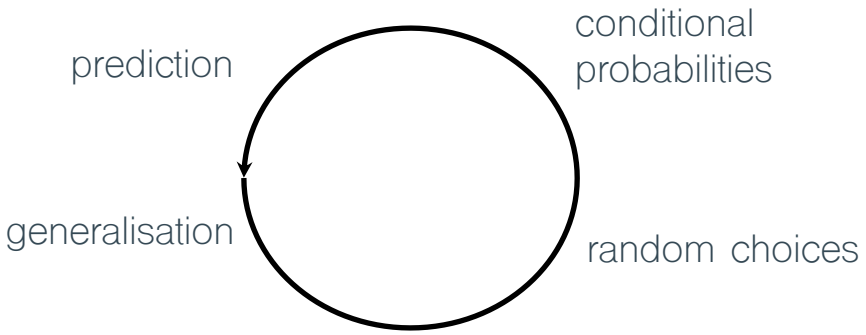
A few insights

1) Given a batch of sentences $\boldsymbol{x}_i$, we repeat the previous process, predicting masked tokens. We wanted to minimise the prediction error. That is, we compare our predictions with the true masked tokens.

2) A relatively large number of tools are used to set up a masked input, which in the end outputs the predicted tokens. For instance in BERT, these ones include an embedding of the NLP tokens into a real space, an encoder to a latent representation and a multi-head transformer, among others.

3) The number of masked tokens is an hyperparameter, which so far has been usually set up to 15-20%. Why that amount is important? I will make some comments on this later.

An early observation



A **naive statistician** could think that we are talking about **Bayesian inference**

Formal results

First of all – Which marginal likelihood?

We consider a latent variable model, where observations are iid

$$p_\theta(\boldsymbol{x}_i) = \int p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)\mathrm{d}\boldsymbol{z}_i$$

$$\log p_\theta(\boldsymbol{x}_{1:n}) = \sum_{i=1}^{n} \log p_\theta(\boldsymbol{x}_i)$$

Formal results

First of all – Which marginal likelihood?

We consider a latent variable model, where observations are iid

$$p_\theta(\boldsymbol{x}_i) = \int p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)\mathrm{d}\boldsymbol{z}_i$$

$$\log p_\theta(\boldsymbol{x}_{1:n}) = \sum_{i=1}^{n} \log p_\theta(\boldsymbol{x}_i)$$

Our goal

Finding a formal connection between the two probabilities

$$\log p_\theta(\boldsymbol{x}_\mathcal{M}|\boldsymbol{x}_\mathcal{R}) = \sum_{t=1}^{M} \log p_\theta(x_{\mathcal{M}(t)}|\boldsymbol{x}_\mathcal{R}) \qquad \log p_\theta(\boldsymbol{x}_i)$$

Formal results

Step 1 – How does the marginal likelihood factorize?

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_1, x_2, x_3, \ldots, x_D)$$

Formal results

Step 1 – How does the marginal likelihood factorize?

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &= \log p_\theta(x_1, x_2, x_3, \ldots, x_D) \\ &= \log p_\theta(x_1 | x_2, x_3, \ldots, x_D) + \log p_\theta(x_2, x_3, \ldots, x_D)\end{aligned}$$

Formal results

Step 1 – How does the marginal likelihood factorize?

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \log p_\theta(x_1, x_2, x_3, \ldots, x_D) \\
&= \log p_\theta(x_1 | x_2, x_3, \ldots, x_D) + \log p_\theta(x_2, x_3, \ldots, x_D) \\
&= \log p_\theta(x_1 | x_2, x_3, \ldots, x_D) + \log p_\theta(x_2 | x_3, \ldots, x_D) + \log p_\theta(x_3, \ldots, x_D) \\
&= \ldots
\end{aligned}
$$

$$
\log p_\theta(\boldsymbol{x}) = \sum_{t=1}^{D} \log p_\theta \left(x_t | \boldsymbol{x}_{t+1:D} \right)
$$

The marginal likelihood of a datapoint can be expressed as a sum of univariate conditionals

Formal results

Step 2 – Is there a unique order? Or there are many

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_D|x_1,\dots) + \log p_\theta(x_{D-1}|x_1,\dots) + \cdots + \log p_\theta(x_2|x_1)$$

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_2|x_1,\dots) + \log p_\theta(x_3|x_1,\dots) + \cdots + \log p_\theta(x_D|x_1)$$

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_2|x_1,\dots) + \log p_\theta(x_4|x_1,\dots) + \cdots + \log p_\theta(x_D|x_1)$$

$$\vdots$$

Formal results

Step 2 – Is there a unique order? Or there are many

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_D|x_1,\dots) + \log p_\theta(x_{D-1}|x_1,\dots) + \cdots + \log p_\theta(x_2|x_1)$$
$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_2|x_1,\dots) + \log p_\theta(x_3|x_1,\dots) + \cdots + \log p_\theta(x_D|x_1)$$
$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_2|x_1,\dots) + \log p_\theta(x_4|x_1,\dots) + \cdots + \log p_\theta(x_D|x_1)$$
$$\vdots$$

Before – One single order

$$\log p_\theta(\boldsymbol{x}) = \sum_{t=1}^{D} \log p_\theta\left(x_t|\boldsymbol{x}_{t+1:D}\right)$$

Now - Averaging over all

$$\log p_\theta(\boldsymbol{x}) = \frac{1}{D!}\sum_{\pi=1}^{D!}\sum_{t=1}^{D} \log p_\theta\left(x^{(\pi)}_{\mathcal{M}(t)}|\boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)}\right)$$

Formal results

Step 3 – Fix the index, re-factorize again

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &= \log p_\theta(x_1, x_2, x_3, \ldots, x_D) \\ &= \log p_\theta(x_1 | x_2, x_3, \ldots, x_D) + \ldots\end{aligned}$$

Formal results

Step 3 – Fix the index, re-factorize again

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta(x_1, x_2, x_3, \ldots, x_D)$$

$$= \log p_\theta(x_1 | x_2, x_3, \ldots, x_D) + \ldots$$

How many choices for the order of conditionals given the rest of tokens?

$$+ \log p_\theta(x_2 | x_3, \ldots, x_D) + \ldots$$

$$+ \log p_\theta(x_3 | x_2, \ldots, x_D) + \ldots$$

$$+ \log p_\theta(x_D | x_2, \ldots, x_{D-1}) + \ldots$$

Formal results

Step 3 – Fix the index, re-factorize again

$$\log p_\theta(\boldsymbol{x}) = \frac{1}{D!} \sum_{\pi=1}^{D!} \sum_{t=1}^{D} \log p_\theta \left(x^{(\pi)}_{\mathcal{M}(t)} | \boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)} \right)$$

$$\sum_{\pi=1}^{D!} \log p_\theta \left(x^{(\pi)}_{\mathcal{M}(t)} | \boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)} \right) = \sum_{\pi=1}^{\mathcal{C}_t} \sum_{j=1}^{D-t+1} \log p_\theta \left(x^{(\pi)}_{\mathcal{M}(j)} | \boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)} \right)$$

where $\mathcal{C}_t \equiv \binom{D}{t-1}$

Formal results

Final step – Putting all together

Using the fact that we are already averaging sums

$$\log p_\theta(\boldsymbol{x}) = \frac{1}{D!}\sum_{\pi=1}^{D!}\sum_{t=1}^{D}\log p_\theta\left(x^{(\pi)}_{\mathcal{M}(t)}|\boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)}\right)$$

We can also sum over all averages

$$\log p_\theta(\boldsymbol{x}) = \sum_{t=1}^{D}\frac{1}{\mathcal{C}_t}\sum_{\pi=1}^{\mathcal{C}_t}\frac{1}{D-t+1}\sum_{j=1}^{D-t+1}\log p_\theta\left(x^{(\pi)}_{\mathcal{M}(j)}|\boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-t)}\right)$$

Formal results

Final step – Putting all together

Using the fact that we are already averaging sums

$$\log p_\theta(\boldsymbol{x}) = \frac{1}{D!}\sum_{\pi=1}^{D!}\sum_{t=1}^{D}\log p_\theta\left(x^{(\pi)}_{\mathcal{M}(t)}|\boldsymbol{x}^{(\pi)}_{\mathcal{M}(t+1:D)}\right)$$

We can also sum over all averages

$$\log p_\theta(\boldsymbol{x}) = \sum_{t=1}^{D}\frac{1}{\mathcal{C}_t}\sum_{\pi=1}^{\mathcal{C}_t}\frac{1}{D-t+1}\sum_{j=1}^{D-t+1}\log p_\theta\left(x^{(\pi)}_{\mathcal{M}(j)}|\boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-t)}\right)$$

Here, we already can spot the pattern

Formal results

Final step – Putting all together

Proposition 1 — *The cumulative expected loss of masked pre-training along the sizes of the mask of tokens $M \in \{1, 2, \dots, D\}$ is equivalent to the log-marginal likelihood of the model when using self-predictive conditionals probabilities, such that*

$$\log p_\theta(\boldsymbol{x}) = \sum_{m=1}^{D} \mathcal{S}_\theta(\boldsymbol{x}; m),$$

where the score function $\mathcal{S}_\theta(\cdot; M)$ corresponds to

$$\mathcal{S}_\theta(\boldsymbol{x}; M) = \frac{1}{\mathcal{C}_M} \sum_{\pi=1}^{\mathcal{C}_M} \frac{1}{M} \sum_{j=1}^{M} \log p_\theta(x^{(\pi)}_{\mathcal{M}(j)} | \boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-j)}) = \frac{1}{M} \mathbb{E}_{\mathcal{M}} \left[\sum_{j=1}^{M} \log p_\theta(x_{\mathcal{M}(j)} | \boldsymbol{x}_{\mathcal{R}}) \right].$$

Proof: In the supplementary material.

MPT on tractable models

The goal now is to apply masked pre-training on tractable models, such that we can understand what is going on. In particular we are interested in the understanding on why this type of learning leads to such good results close to proper generalisation performance (at least on NLP).

Probabilistic PCA as the *proof-of-concept*

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \epsilon,$$

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W}, \boldsymbol{\mu}, \sigma_0^2) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma_0^2\mathbb{I}), \qquad \text{where } \epsilon \sim \mathcal{N}(0, \sigma_0^2\mathbb{I}),\ \boldsymbol{\mu} \in \mathbb{R}^D \text{ and } \boldsymbol{W} \in \mathbb{R}^{D\times K}.$$

MPT on tractable models

The goal now is to apply masked pre-training on tractable models, such that we can understand what is going on. In particular we are interested in the understanding on why this type of learning leads to such good results close to proper generalisation performance (at least on NLP).

Probabilistic PCA as the *proof-of-concept*

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \epsilon,$$

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W}, \boldsymbol{\mu}, \sigma_0^2) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma_0^2\mathbb{I}), \qquad \text{where } \epsilon \sim \mathcal{N}(0, \sigma_0^2\mathbb{I}),\ \boldsymbol{\mu} \in \mathbb{R}^D \text{ and } \boldsymbol{W} \in \mathbb{R}^{D\times K}.$$

$$p_\theta(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{W}, \boldsymbol{\mu}, \sigma_0^2) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^\top + \sigma_0^2\mathbb{I}),$$

$$p_\theta(\boldsymbol{x}_{1:n}) = \prod_{i=1}^{n} p_\theta(\boldsymbol{x}_i).$$

where we used $\theta = \{\boldsymbol{W}, \boldsymbol{\mu}, \sigma_0^2\}$.

MPT on tractable models

Probabilistic PCA

We can now easily compute log-marginal likelihood (evidence) and compare if MPT is close enough to it. At least, that could indicate what are we indeed maximising under this new way of learning.

```
74  ########################################
75  # EXACT LOG-MARGINAL LIKELIHOOD
76  ########################################
77  S_lml = W @ W.T + (1/beta)*torch.eye(D)
78  lml_dist = Normal(torch.zeros(D), S_lml)
79  lml = lml_dist.log_prob(x).sum()
```

MPT on tractable models

Probabilistic PCA

We can now easily compute log-marginal likelihood (evidence) and compare if MPT is close enough to it. At least, that could indicate what are we indeed maximising under this new way of learning.

```
74  ########################################
75  # EXACT LOG-MARGINAL LIKELIHOOD
76  ########################################
77  S_lml = W @ W.T + (1/beta)*torch.eye(D)
78  lml_dist = Normal(torch.zeros(D), S_lml)
79  lml = lml_dist.log_prob(x).sum()
```

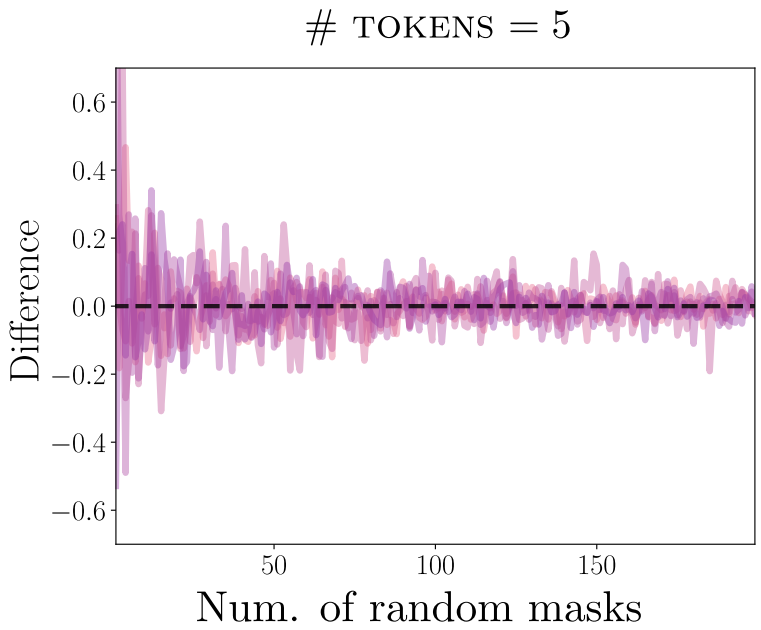
Posterior predictive probabilities. The predictive distribution between the dimensions of $\boldsymbol{x}_i$ can be obtained from both latent variable integration or by properties of Gaussian conditionals. In our case, we use the latter example. Thus, having both *mask* $\mathcal{M}$ and *rest* $\mathcal{R}$ indices according to our previous notation, we can look to the multivariate normal distribution $p_\theta(\boldsymbol{x})$ using *block* submatrices, such that

$$p_\theta(\boldsymbol{x}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_\mathcal{M}\\ \boldsymbol{x}_\mathcal{R}\end{bmatrix} \middle| \begin{bmatrix}\boldsymbol{\mu}_\mathcal{M}\\ \boldsymbol{\mu}_\mathcal{R}\end{bmatrix}, \begin{bmatrix}\boldsymbol{S}_{\mathcal{M}\mathcal{M}} & \boldsymbol{S}_{\mathcal{M}\mathcal{R}}\\ \boldsymbol{S}_{\mathcal{M}\mathcal{R}}^\top & \boldsymbol{S}_{\mathcal{R}\mathcal{R}}\end{bmatrix}\right),$$

where we also defined $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{W}^\top + \sigma_0^2\mathbb{I}$. Using the properties of conditional probabilities on normal distributions, we can write the posterior predictive densities in closed-form, such that $p_\theta(\boldsymbol{x}_\mathcal{M}|\boldsymbol{x}_\mathcal{R}) = \mathcal{N}(\boldsymbol{m}_{\mathcal{M}|\mathcal{R}}, \boldsymbol{v}_{\mathcal{M}|\mathcal{R}})$, where parameters are obtained from

$$\boldsymbol{m}_{\mathcal{M}|\mathcal{R}} = \boldsymbol{\mu}_\mathcal{M} + \boldsymbol{S}_{\mathcal{M}\mathcal{R}}^\top \boldsymbol{S}_{\mathcal{R}\mathcal{R}}^{-1}(\boldsymbol{x}_\mathcal{R} - \boldsymbol{\mu}_\mathcal{R}), \quad \boldsymbol{v}_{\mathcal{M}|\mathcal{R}} = \boldsymbol{S}_{\mathcal{M}\mathcal{M}} + \boldsymbol{S}_{\mathcal{M}\mathcal{R}}^\top \boldsymbol{S}_{\mathcal{R}\mathcal{R}}^{-1}\boldsymbol{S}_{\mathcal{M}\mathcal{R}}.$$

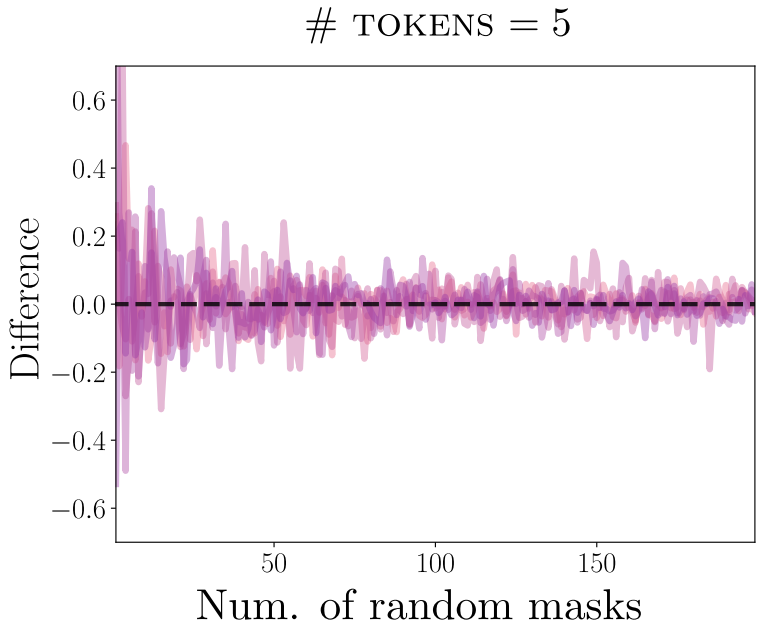
Results: **Convergence**



Main points to check in these empirical results

1) Does it converge to the true log-marginal likelihood?
2) How fast does it converge?

Results: Convergence



Proposition 1 — *The cumulative expected loss of masked pre-training along the sizes of the mask of tokens $M \in \{1, 2, \ldots, D\}$ is equivalent to the log-marginal likelihood of the model when using self-predictive conditionals probabilities, such that*

$$\log p_\theta(\boldsymbol{x}) = \sum_{m=1}^{D} \mathcal{S}_\theta(\boldsymbol{x}; m),$$

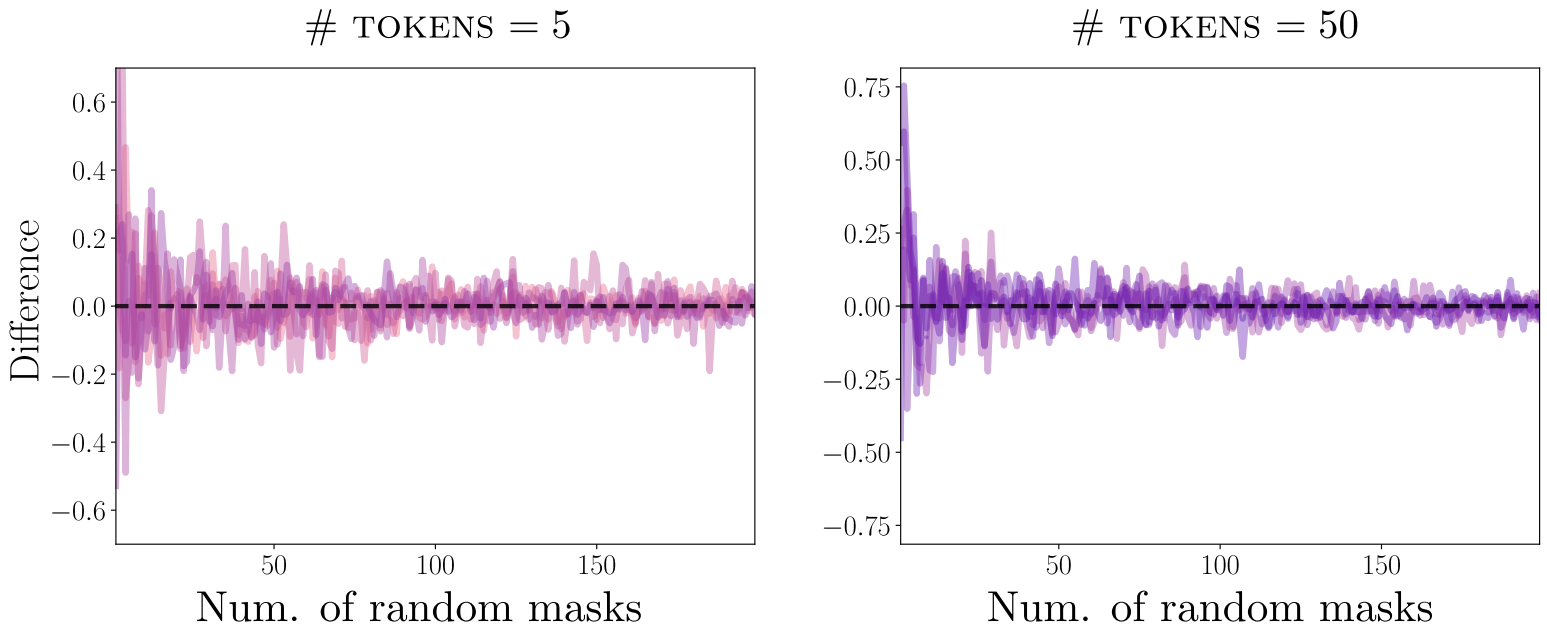
where the score function $\mathcal{S}_\theta(\cdot; M)$ corresponds to

$$\mathcal{S}_\theta(\boldsymbol{x}; M) = \frac{1}{\mathcal{C}_M} \sum_{\pi=1}^{\mathcal{C}_M} \frac{1}{M} \sum_{j=1}^{M} \log p_\theta(x^{(\pi)}_{\mathcal{M}(j)} | \boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-j)}) = \frac{1}{M} \mathbb{E}_{\mathcal{M}} \left[\sum_{j=1}^{M} \log p_\theta(x_{\mathcal{M}(j)} | \boldsymbol{x}_{\mathcal{R}}) \right].$$

Proof: In the supplementary material.

Main points to check in these empirical results
1) Does it converge to the true log-marginal likelihood?
2) How fast does it converge?

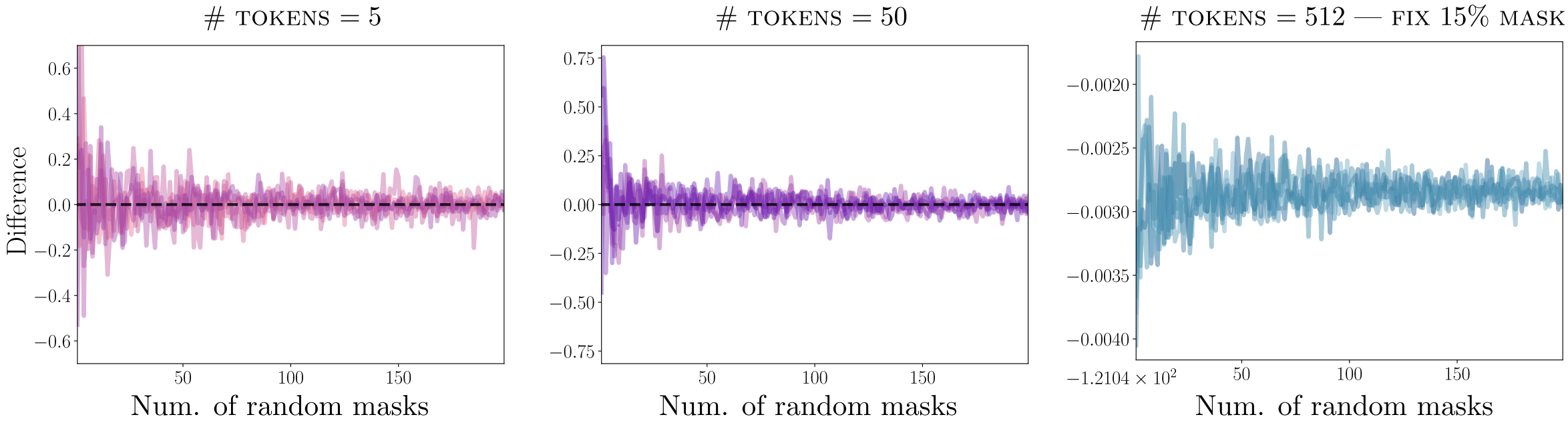
Results: **Convergence**



Main points to check in these empirical results

1) Does it converge to the true log-marginal likelihood?
2) How fast does it converge?

Results: **Convergence**



Main points to check in these empirical results

1) Does it converge to the true log-marginal likelihood?
2) How fast does it converge?
3) What if we do not try all different mask sizes?

Results: Convergence

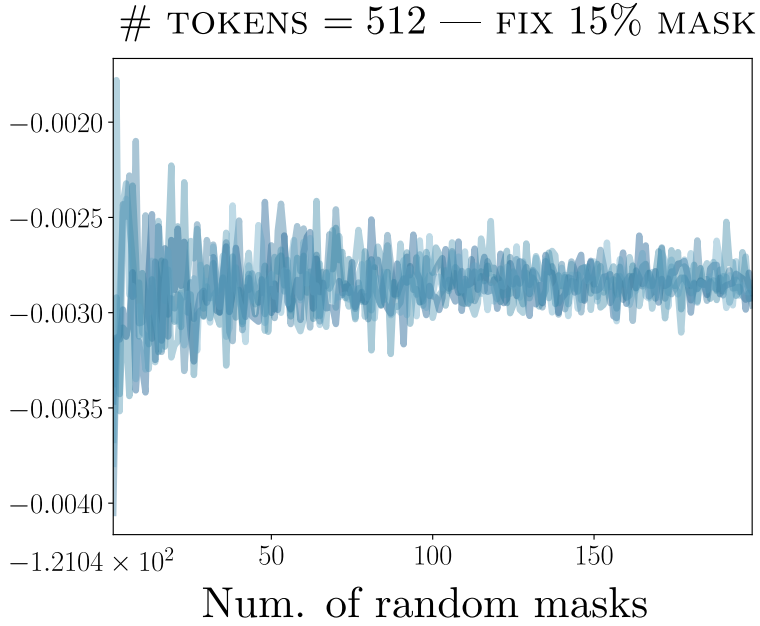
Proposition 1 — *The cumulative expected loss of masked pre-training along the sizes of the mask of tokens $M \in \{1, 2, \dots, D\}$ is equivalent to the log-marginal likelihood of the model when using self-predictive conditionals probabilities, such that*

$$\log p_\theta(\boldsymbol{x}) = \sum_{m=1}^{D} \mathcal{S}_\theta(\boldsymbol{x}; m),$$

where the score function $\mathcal{S}_\theta(\cdot; M)$ corresponds to

$$\mathcal{S}_\theta(\boldsymbol{x}; M) = \frac{1}{C_M} \sum_{\pi=1}^{C_M} \frac{1}{M} \sum_{j=1}^{M} \log p_\theta(x^{(\pi)}_{\mathcal{M}(j)} | \boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-j)}) = \frac{1}{M} \mathbb{E}_{\mathcal{M}} \left[\sum_{j=1}^{M} \log p_\theta(x_{\mathcal{M}(j)} | \boldsymbol{x}_{\mathcal{R}}) \right].$$

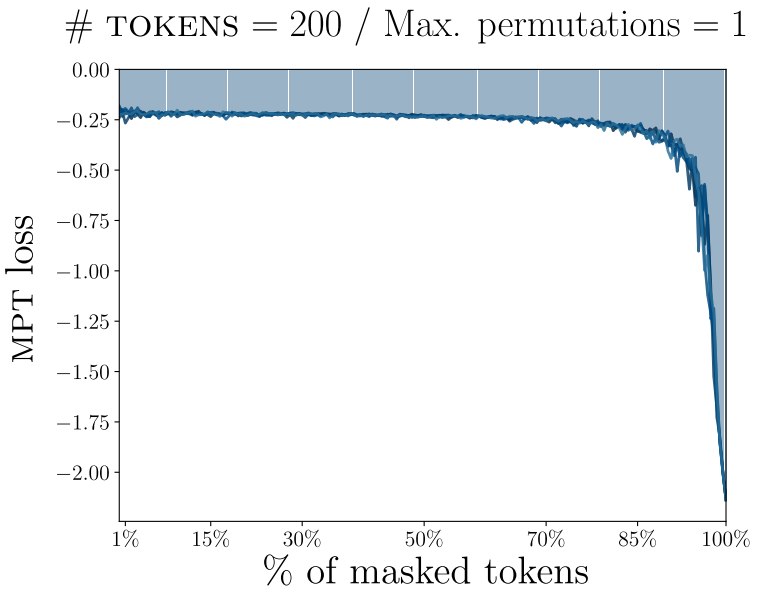
Proof: In the supplementary material.



Main points to check in these empirical results

1) Does it converge to the true log-marginal likelihood?
2) How fast does it converge?
3) What if we do not try all different mask sizes?

Results: **Cumulative sums**



Proposition 1 — *The cumulative expected loss of masked pre-training along the sizes of the mask of tokens $M \in \{1, 2, \dots, D\}$ is equivalent to the log-marginal likelihood of the model when using self-predictive conditionals probabilities, such that*

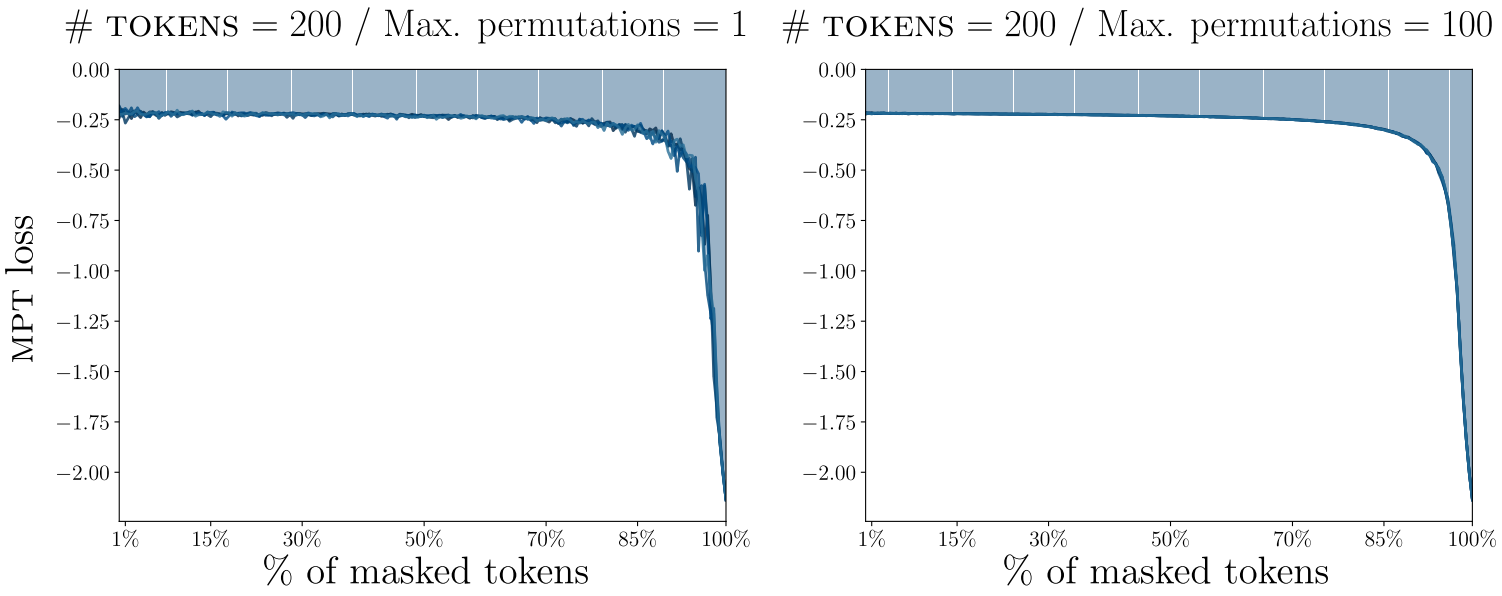
$$\log p_\theta(\boldsymbol{x}) = \sum_{m=1}^{D} \mathcal{S}_\theta(\boldsymbol{x}; m),$$

where the score function $\mathcal{S}_\theta(\cdot; M)$ corresponds to

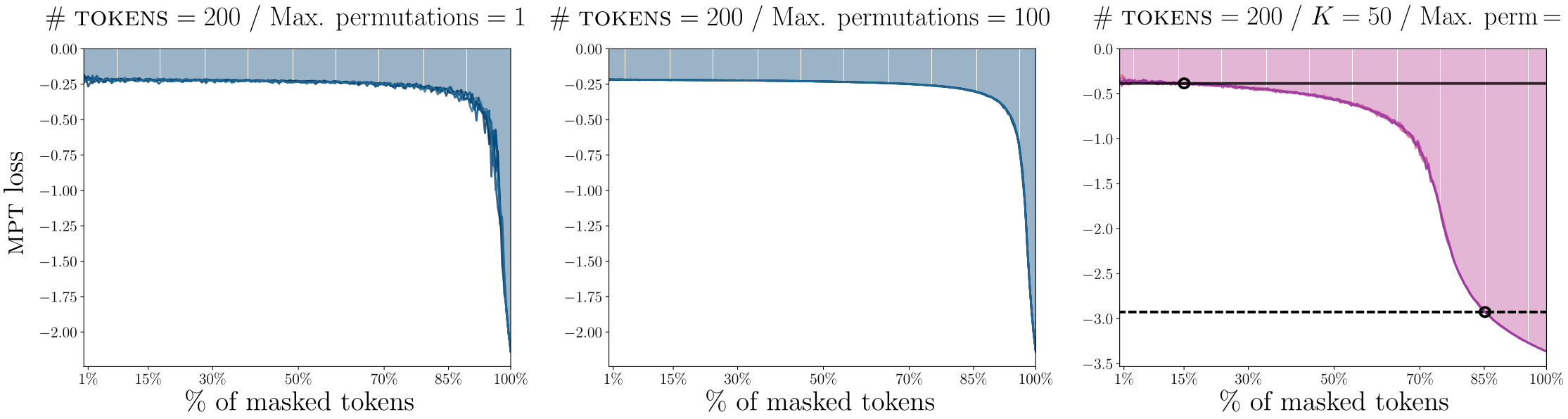
$$\mathcal{S}_\theta(\boldsymbol{x}; M) = \frac{1}{C_M} \sum_{\pi=1}^{C_M} \frac{1}{M} \sum_{j=1}^{M} \log p_\theta(x^{(\pi)}_{\mathcal{M}(j)} | \boldsymbol{x}^{(\pi)}_{\mathcal{R}(1:D-j)}) = \frac{1}{M} \mathbb{E}_{\mathcal{M}} \left[\sum_{j=1}^{M} \log p_\theta(x_{\mathcal{M}(j)} | \boldsymbol{x}_{\mathcal{R}}) \right].$$

Proof: In the supplementary material.

Results: **Cumulative sums**

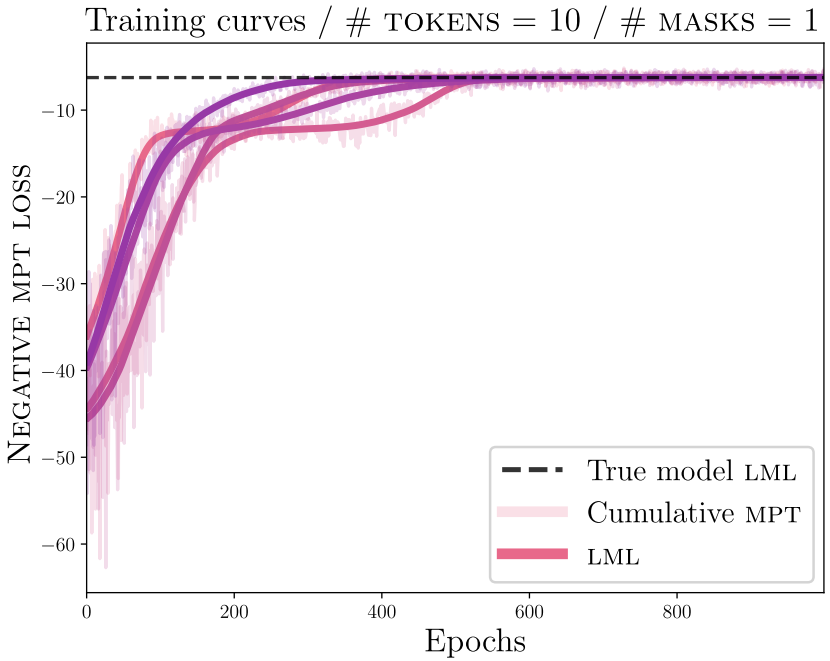


Results: **Cumulative sums**



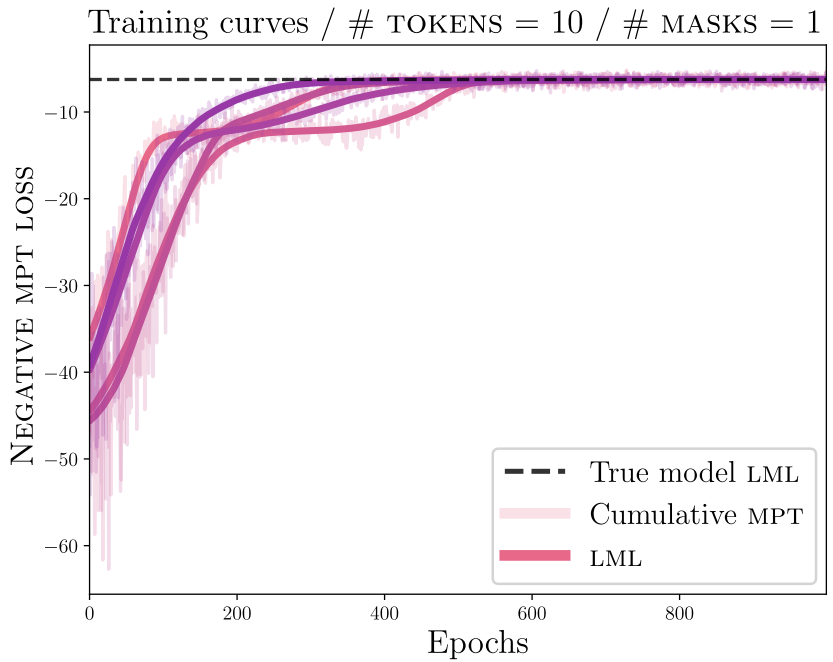
What are we then doing when fixing the mask ratio?

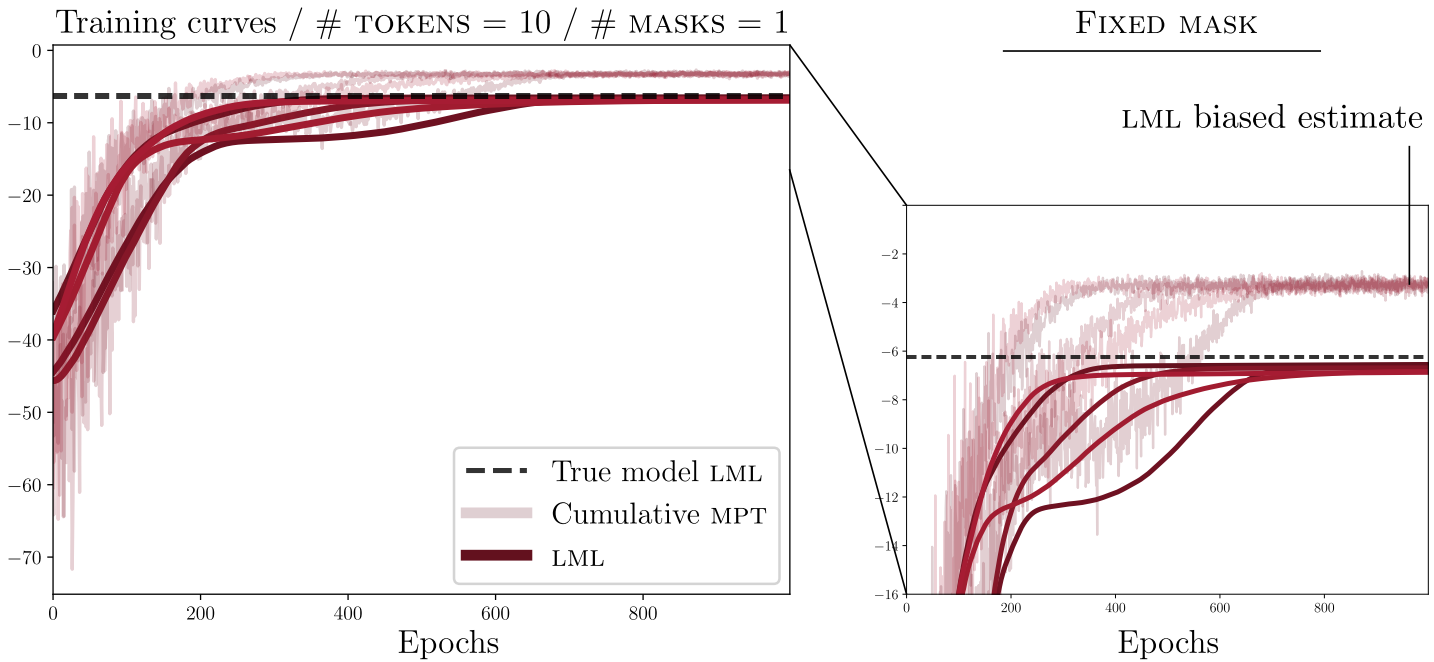
Results: Training



Being rigorous wrt the main results leads us to the true marginal likelihood

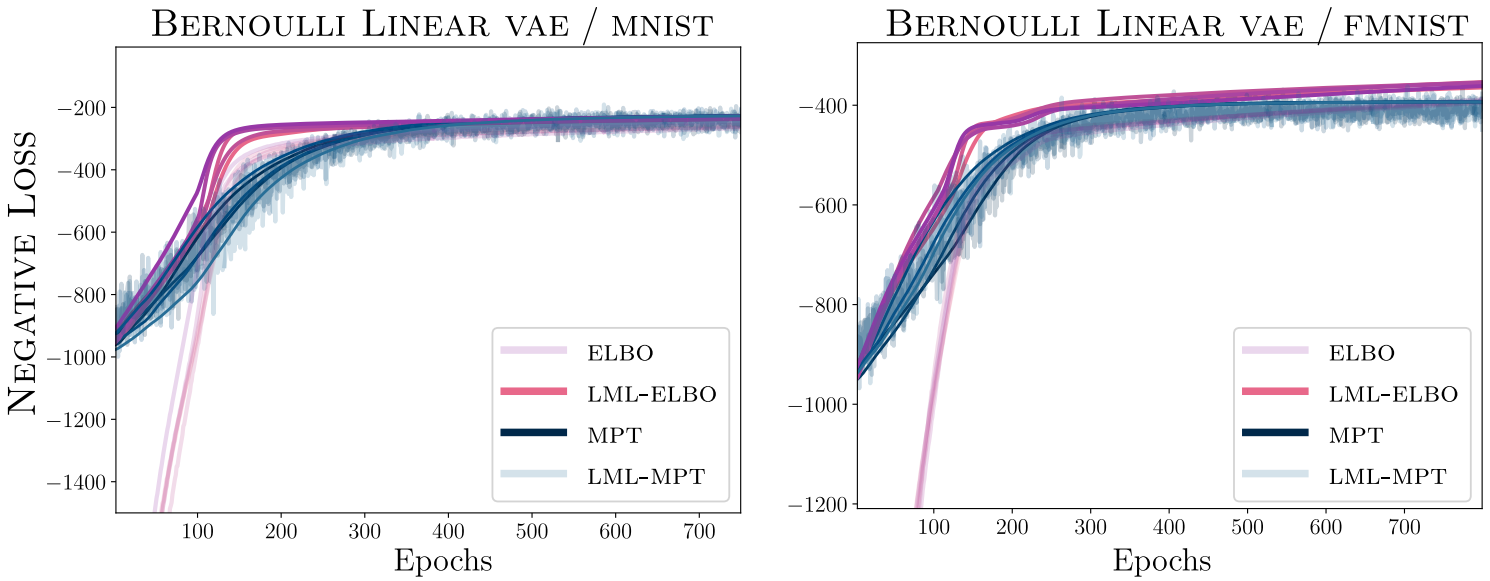
Results: Training



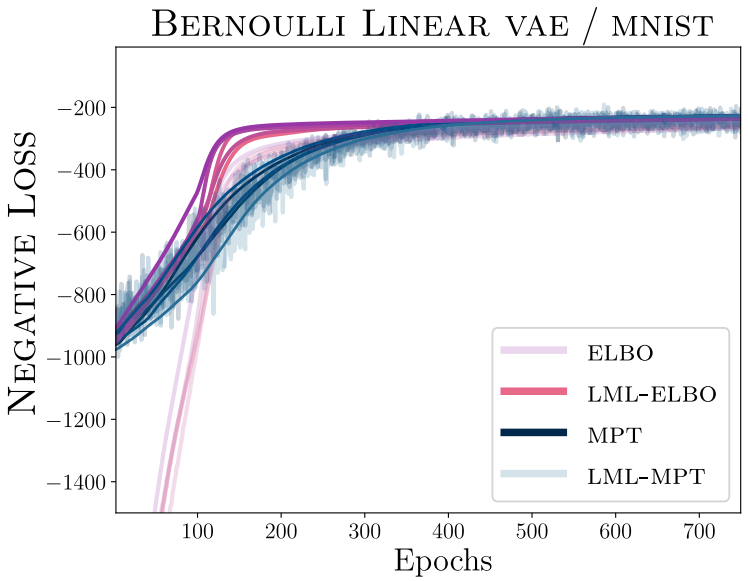


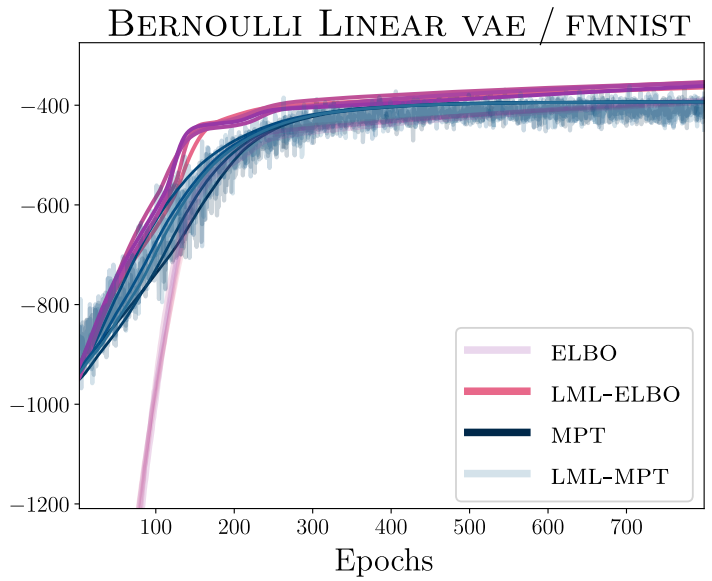
Biased estimation also maximises marginal likelihood

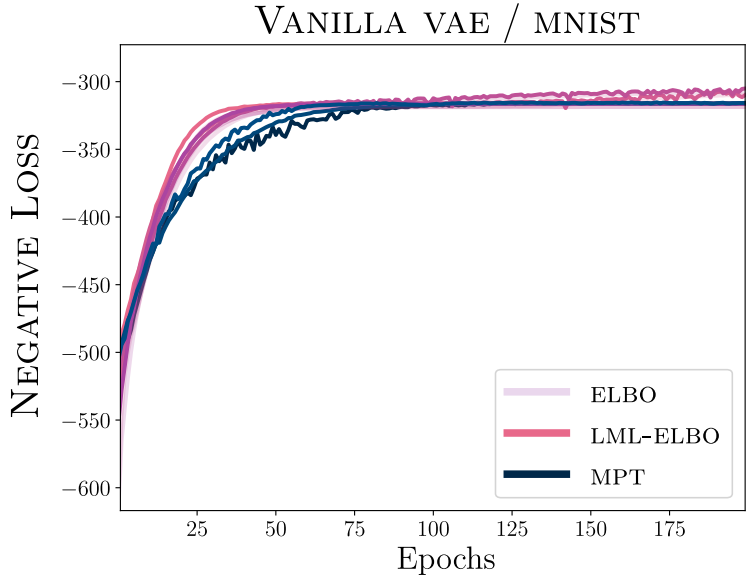
Results: Training



Results: Training

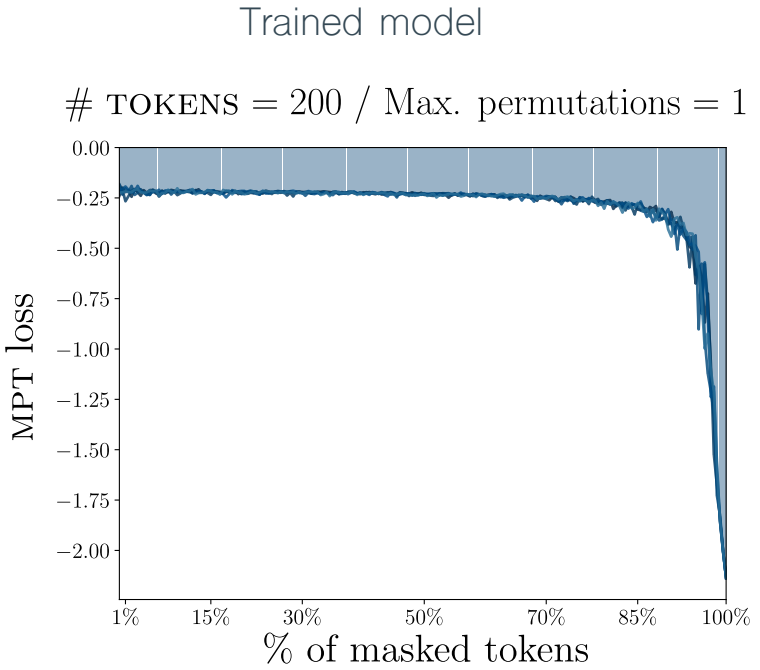




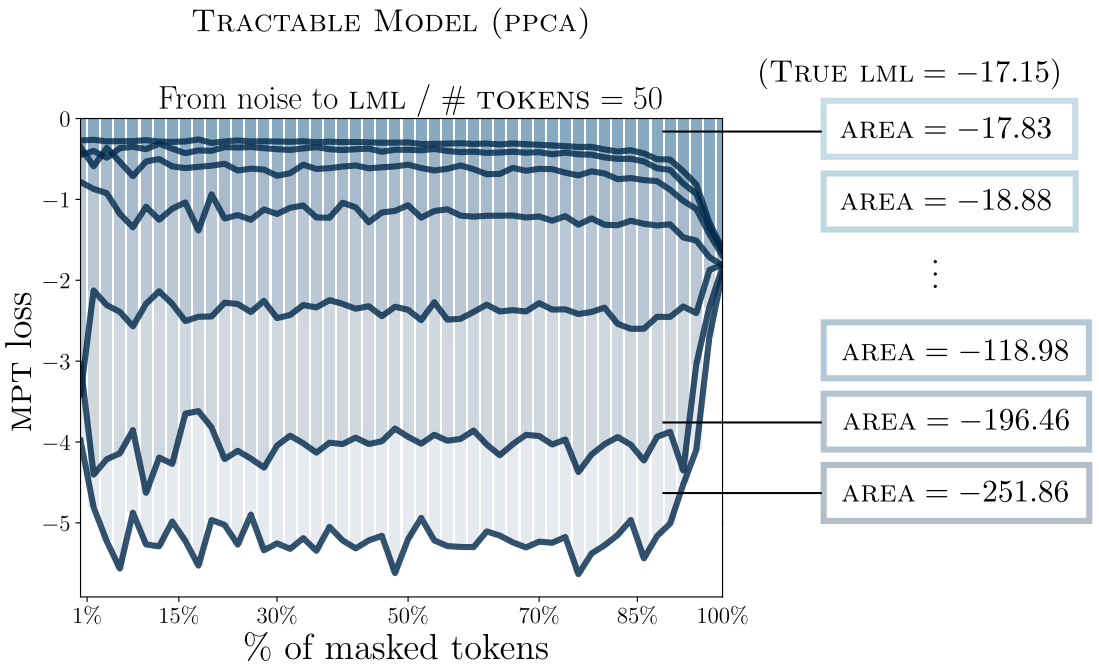


Predictive conditionals are not trivial with VAEs

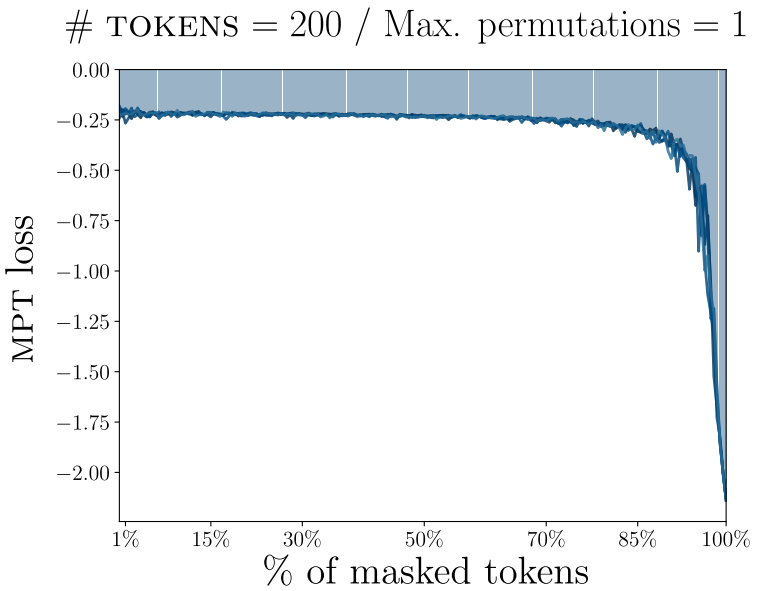
Results: **BERT**



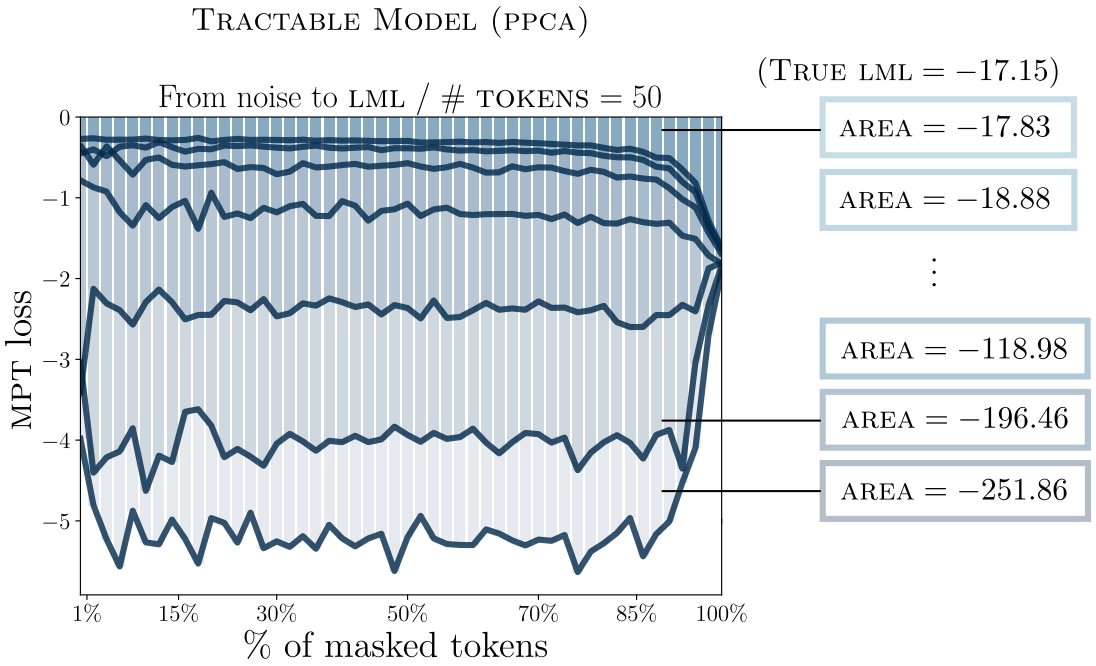
Results: **BERT**

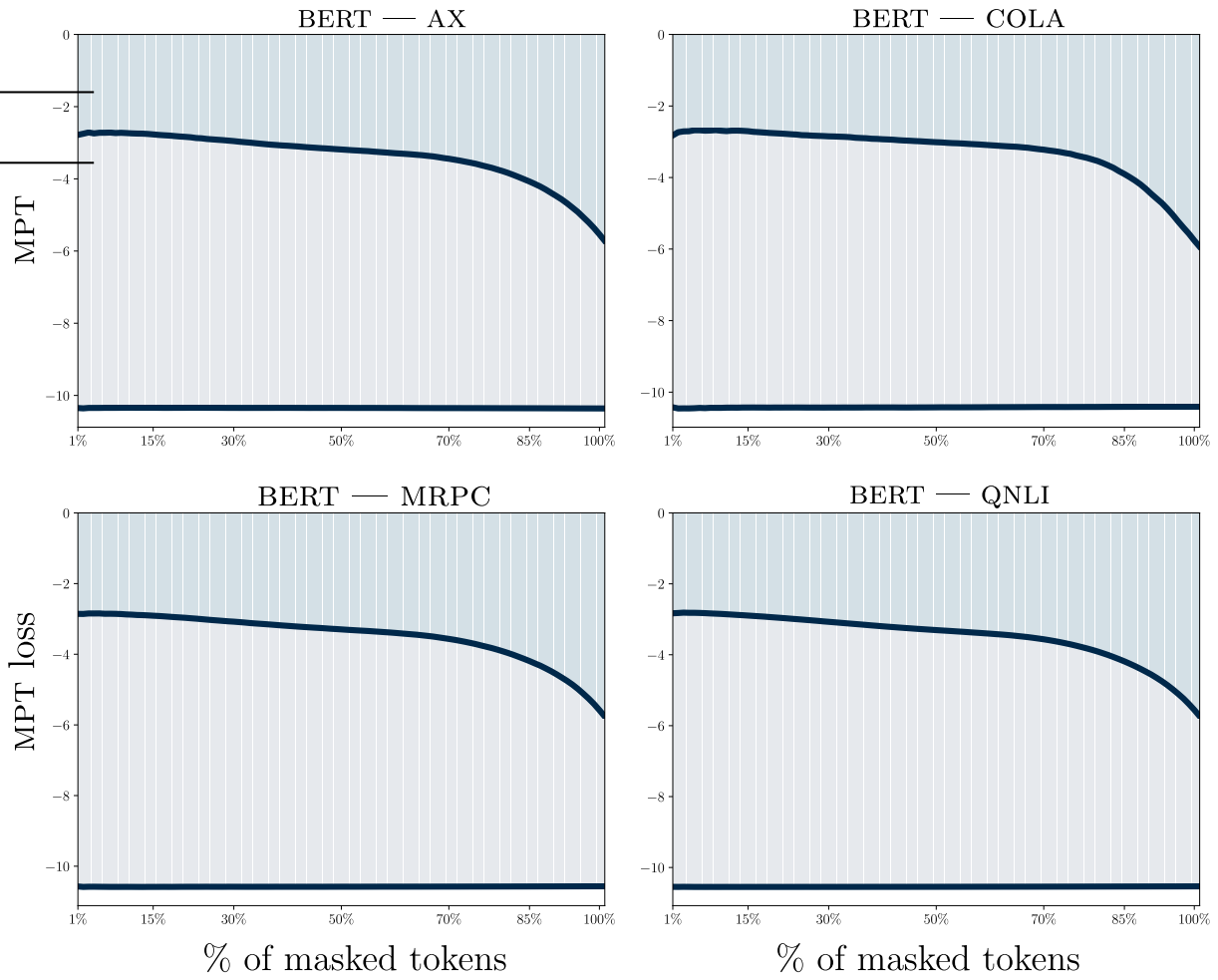


Trained model

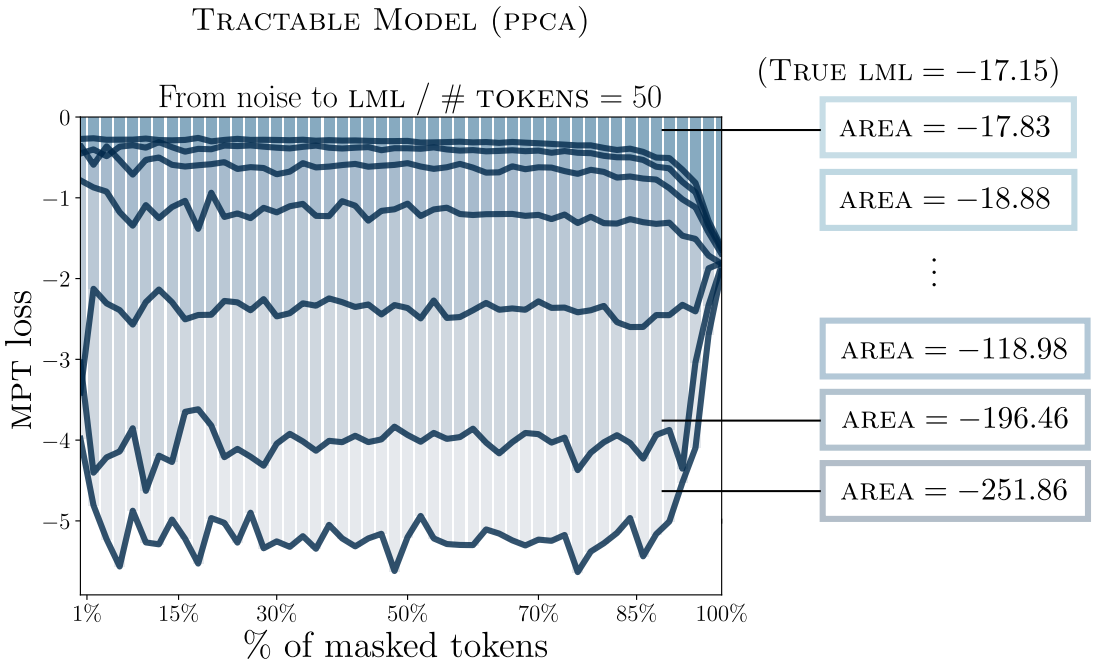


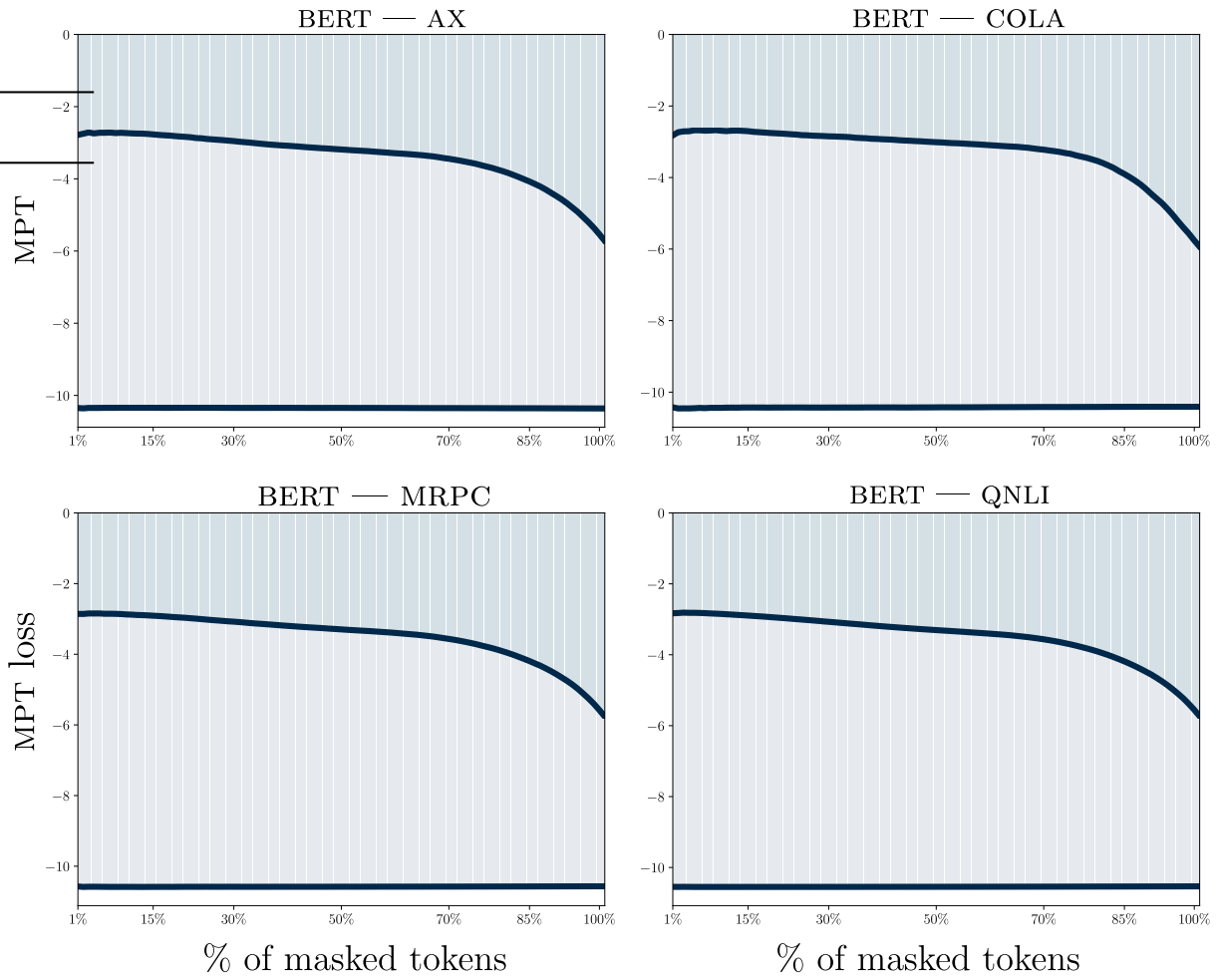
Results: BERT





Results: BERT





Conclusions

1) We confirm that MPT maximizes according to a stochastic gradient of the log-marginal likelihood of the model.

Conclusions

1) We confirm that MPT maximizes according to a stochastic gradient of the log-marginal likelihood of the model.

2) We understand the underlying effect of the masking ratio, at least in terms of the biased estimation and the area-under-the-curve. An hypothesis about having lower or upper bound dependent on the masked ratio could fit. Do we have a tighter bound/estimator with 25% instead of 15%?

Conclusions

1) We confirm that MPT maximizes according to a stochastic gradient of the log-marginal likelihood of the model.

2) We understand the underlying effect of the masking ratio, at least in terms of the biased estimation and the area-under-the-curve. An hypothesis about having lower or upper bound dependent on the masked ratio could fit. Do we have a tighter bound/estimator with 25% instead of 15%?

3) Empirical results match the theory very well – Conditional probabilities seems to be a powerful tool that converges to desired measures of generalization. Bayesian modelling could benefit from these.

Collaborators

Pol G. Recasens
(soon) PhD student
@ Barcelona Supercomputing Center (BSC-CNS)

Søren Hauberg
Professor
@ Technical University of Denmark (DTU)

Thanks!

Preprint is currently available –

`https://arxiv.org/abs/2306.00520`

Code for reproducing results –

`https://github.com/pmorenoz/MPT-LML`