K-priors: A General Principle of Adaptation

Mohammad Emtiyaz Khan
RIKEN Center for AI Project, Tokyo
http://emtiyaz.github.io
Continual Learning: Lifelong and incremental

 Quickly adapt to new situations by exploiting (and preserving) the past knowledge

Human Learning at the age of 6 months.
Human Learning at the age of 6 months.
Human Learning at the age of 6 months.
Converged at the age of 12 months
Converged at the age of 12 months
Converged at the age of 12 months
Transfer skills at the age of 14 months
Transfer skills at the age of 14 months
Transfer skills at the age of 14 months
Adaptation in Machine Learning

3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s
Adaptation in Machine Learning

• Changes in the training frameworks [1,2]
  – New data are regularly pooled and labeled
  – Old data become irrelevant
  – Regular hyperparameter tuning to handle drifts
  – Model class/architectures needs an update

3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s
Adaptation in Machine Learning

• Changes in the training frameworks [1,2]
  – New data are regularly pooled and labeled
  – Old data become irrelevant
  – Regular hyperparameter tuning to handle drifts
  – Model class/architectures needs an update
• Constant retraining, retesting, redeployment
  – Huge financial and environmental costs (e.g., Tesla AI DataEngine takes 70000 GPU hrs [3])

3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s
Failure of AI in “dynamic” setting

Microsoft’s chatbot “Tay Tweets” went crazy only after 24 hours of “learning” from the other people’s tweets (2016)

http://smerity.com/articles/2016/tayandyou.html
Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed (for example, at homes for elderly care)

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017
Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed (for example, at homes for elderly care)

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017
This Talk

This Talk

• Adaptation mechanisms that are
  – Quick (avoid full retraining)
  – Accurate (performance similar to retraining)
  – Wide (works for variety of tasks and models)

This Talk

• Adaptation mechanisms that are
  – Quick (avoid full retraining)
  – Accurate (performance similar to retraining)
  – Wide (works for variety of tasks and models)

• Knowledge-Adaptation priors (K-priors) [1]
  – Principle: reconstruct the gradient of the “past”
  – Unify & generalize many adaptation strategies (weight priors, knowledge distillation, similarity control, SVMs, GPs, and memory-based CL)

Knowledge-Adaptation Priors

Mohammad Emtiyaz Khan*  
RIKEN Center for AI Project  
Tokyo, Japan  
emtiyaz.khan@riken.jp

Siddharth Swaroop*  
University of Cambridge  
Cambridge, UK  
ss2163@cam.ac.uk

Abstract

Humans and animals have a natural ability to quickly adapt to their surroundings, but machine-learning models, when subjected to changes, often require a complete retraining from scratch. We present Knowledge-adaptation priors (K-priors) to reduce the cost of retraining by enabling quick and accurate adaptation for a wide-variety of tasks and models. This is made possible by a combination of weight and function-space priors to reconstruct the gradients of the past, which recovers and generalizes many existing, but seemingly-unrelated, adaptation strategies. Training with simple first-order gradient methods can often recover the exact retrained model to an arbitrary accuracy by choosing a sufficiently large memory of the past data. Empirical results confirm that the adaptation can be cheap and accurate, and a promising alternative to retraining.

Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

$$\sum_{i \in D} \ell_i(w) + \mathcal{R}(w)$$
Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

$$\ell_j(w) + \sum_{i \in D} \ell_i(w) + \mathcal{R}(w)$$

Add data
Adaptation Tasks

Given a base model $w_*$ trained on data $\mathcal{D}$, adapt it to “incremental” changes in the training framework

$$-l_k(w) + l_j(w) + \sum_{i \in \mathcal{D}} l_i(w) + \mathcal{R}(w)$$

Delete data Add data
Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

\[-l_k(w) + l_j(w) + \sum_{i \in D} l_i(w) + R(w) - R(w) + G(w)\]

Delete data \hspace{1cm} Add data \hspace{1cm} Change regularizer or hyperparameter
Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

\[-l_k(w) + l_j(w) + \sum_{i \in D} l_i(w) + R(w) - R(w) + G(w)\]

Delete data  Add data  Change regularizer or hyperparameter

Change model $f^i_w$ or architecture
Speeding up K-fold Cross-Validation

Every run in CV can be “quickly adapted” using the model trained in the previous run [1]

Speeding up K-fold Cross-Validation

Every run in CV can be “quickly adapted” using the model trained in the previous run [1]

Speeding up K-fold Cross-Validation

Every run in CV can be “quickly adapted” using the model trained in the previous run [1]

Speeding up K-fold Cross-Validation

Every run in CV can be “quickly adapted” using the model trained in the previous run [1]

Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

$$-l_k(w) + l_j(w) + \sum_{i \in D} l_i(w) + R(w) - R(w) + G(w)$$

- Delete data
- Add data
- Change model $f_w^i$ or architecture
- Change regularizer or hyperparameter
Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

$$-l_k(w) + l_j(w) + \sum_{i \in D} l_i(w) + R(w) - R(w) + G(w)$$

Delete data  Add data  Change model $f_w^i$ or architecture

Change regularizer or hyperparameter

Adaptation mechanisms that are accurate, quick, work for all these tasks, and for generic model $f_w^i$. 
Adaptation Tasks

Given a base model $w_*$ trained on data $D$, adapt it to “incremental” changes in the training framework

\[
-l_k(w) + l_j(w) + \sum_{i \in D} \ell_i(w) + R(w) - R(w) + G(w)
\]

Delete data  Add data  Change model $f_w^i$ or architecture

\[
(w - w_*)^T G(w_*)(w - w_*)
\]

Change regularizer or hyperparameter

Weight-priors
$G$ is Hessian/Fisher [1], Quick, but not wide/accurate

Adaptation mechanisms that are accurate, quick, work for all these tasks, and for generic model $f_w^i$.

Inaccuracy of Weight-Priors


‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.
Inaccuracy of Weight-Priors


‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.
Inaccuracy of Weight-Priors


‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, i.e., decision boundary pass through the origin).

Each task N=500, each class 250 examples.

New data

Weight-prior (bad)

base model

batch training
Inaccuracy of Weight-Priors


‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.
Knowledge-Adaptation Priors

K-priors use past-memory $\mathcal{M}$ (size M) in addition to the base model.
Knowledge-Adaptation Priors

K-priors use past-memory $\mathcal{M}$ (size M) in addition to the base model.
Knowledge-Adaptation Priors

K-priors use past-memory $\mathcal{M}$ (size M) in addition to the base model.
Knowledge-Adaptation Priors

K-priors use past-memory $\mathcal{M}$ (size M) in addition to the base model.
A General Principle of Adaptation

K-priors $K(w; w_*, M)$ use $w_*$ and $M$

$$-\ell_k(w) + \ell_j(w) + \sum_{i \in D} \ell_i(w) + R(w) - R(w) + G(w)$$
A General Principle of Adaptation

K-priors $K(w; w_*, \mathcal{M})$ use $w_*$ and $\mathcal{M}$

$$-\ell_k(w) + \ell_j(w) + \sum_{i \in \mathcal{D}} \frac{\ell_i(w) + R(w)}{K(w; w_*, \mathcal{M})} - R(w) + G(w)$$
A General Principle of Adaptation

K-priors $K(w; w_*, \mathcal{M})$ use $w_*$ and $\mathcal{M}$

$$-\ell_k(w) + \ell_j(w) + \sum_{i \in \mathcal{D}} \ell_i(w) + R(w) - R(w) + G(w)$$

The principle is to choose $K(w)$ and memory $\mathcal{M}$ s.t. the “gradient of the past” is faithfully reconstructed.

$$\nabla K(w) \approx \nabla \left[ \sum_{i \in \mathcal{D}} \ell_i(w) + R(w) \right]$$
K-prior Construction

Combine weight and function-space divergences

\[ \mathcal{K}(w) = \tau D_w(w \parallel w_\star) + D_f(f(w) \parallel f(w_\star)) \]
K-prior Construction

Combine weight and function-space divergences

\[ \mathcal{K}(w) = \tau \mathbb{D}_w(w \| w_*) + \mathbb{D}_f(f(w) \| f(w_*)) \]
K-prior Construction

Combine weight and function-space divergences

\[ K(w) = \tau \mathcal{D}_w(w \| w_*) + \mathcal{D}_f(f(w) \| f(w_*)) \]

No labels required, so \( \mathcal{M} \) can include any inputs!
Faithful Gradient Reconstruction
Faithful Gradient Reconstruction

K-prior optimum
Faithful Gradient Reconstruction

No labels required, so $M$ can include any inputs!
Faithful Gradient Reconstruction

No labels required, so $\mathcal{M}$ can include any inputs!
Exact Gradient Reconstruction

Consider logistic regression \( f^i_w = \phi_i^T w \)

\[
\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f^i_w)) + \delta\|w\|^2
\]
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

$$K(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Weight-space
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Weight-space

Memory = all past data
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

Weight-space

$$K(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Weight-space

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - \sigma(f_{w*}^i)) + \delta (w - w_*)$$
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in D} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

**Function-space**

**Weight-space**

$$\mathcal{K}(w) = \sum_{i \in D} \ell(\sigma(f_w^i), \sigma(f_w^*)^i) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the **exact** gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in D} \phi_i (\sigma(f_w^i) - \sigma(f_w^*)^i) + \delta (w - w_*)$$
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_w^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Weight-space

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - \sigma(f_w_*^i)) + \delta (w - w_*)$$

$$= \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - y_i) + \delta w - \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w_*^i) - y_i) - \delta w_*$$
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$
\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta\|w\|^2
$$

Function-space

$$
\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_w^i), \sigma(f_w^*) + \delta\|w - w^*\|^2
$$

Weight-space

Memory = all past data

The K-prior recovers the exact gradients!

$$
\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - \sigma(f_w^*)) + \delta(w - w^*)
$$

$$
= \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - y_i) + \delta w - \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^*) - y_i) - \delta w^*
$$

$$
\nabla \bar{l}(w^*) = 0
$$
Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w^*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Function-space
Weight-space

Memory = all past data

The K-prior recovers the **exact** gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - \sigma(f_{w^*}^i)) + \delta (w - w_*)$$

$$= \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - y_i) + \delta w \frac{\sum_{i \in \mathcal{D}} \phi_i (\sigma(f_{w^*}^i) - y_i)}{\delta w_*}$$

$$\nabla \bar{l}(w_*) = 0$$
Exact Gradient Reconstruction

Consider logistic regression $f^i_w = \phi_i^T w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f^i_w)) + \delta \|w\|^2$$

$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f^i_{w*}), \sigma(f^i_w)) + \delta \|w - w_*\|^2$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i(\sigma(f^i_w) - \sigma(f^i_{w*})) + \delta(w - w_*)$$

$$= \sum_{i \in \mathcal{D}} \phi_i(\sigma(f^i_w) - y_i) + \delta w - y_i + y_i$$

$$\nabla \bar{l}(w)$$

$$\sum_{i \in \mathcal{D}} \phi_i(\sigma(f^i_{w*}) - y_i) \delta w_*$$

$$\nabla \bar{l}(w_*) = 0$$
How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

\[
\nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in D \setminus M} \phi_i(\sigma(f^i_w) - \sigma(f^i_{w^*}))
\]

Prediction disagreement
How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

$$\nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in D \setminus M} \phi_i (\sigma(f_w^i) - \sigma(f_{w*}^i))$$

Prediction disagreement

$$\approx \left[ \sum_{i \in D \setminus M} \phi_i \sigma'(f_{w*}^i) \phi_i^\top \right] (w - w_*)$$

2nd derivative of the loss

Generalized Gauss-Newton (GGN)

Independent of $w$
How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

\[ \nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in D \setminus M} \phi_i \left( \sigma(f^i_w) - \sigma(f^i_{w*}) \right) \]

Prediction disagreement

\[ \approx \sum_{i \in D \setminus M} \phi_i \sigma'(f^i_{w*}) \phi_i^T (w - w*) \]

2nd derivative of the loss

Generalized Gauss-Newton (GGN)

Independent of w

Pick points to minimize the GGN approximations. We can use any low-rank approximation. We pick top-M \( \sigma'(f^i_{w*}) \) which is called memorable past \[1\].

---

Memorable Past: Example

base model

batch training

↑ K-prior

Memory(M=4)
Least Memorable

Most Memorable

## Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

<table>
<thead>
<tr>
<th>Method</th>
<th>Wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight priors [1]</td>
<td>✗</td>
</tr>
<tr>
<td>SVMs [2]</td>
<td>✗</td>
</tr>
<tr>
<td>Knowledge Distillation [3]</td>
<td>✗</td>
</tr>
<tr>
<td>Learning under privileged info [4]</td>
<td>✗</td>
</tr>
<tr>
<td>Gaussian Process [5]</td>
<td>✗</td>
</tr>
<tr>
<td>Memory-based CL [6]</td>
<td>✗</td>
</tr>
</tbody>
</table>

---

# Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

<table>
<thead>
<tr>
<th>Method</th>
<th>Wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight priors [1]</td>
<td>✗</td>
</tr>
<tr>
<td>SVMs [2]</td>
<td>✗</td>
</tr>
<tr>
<td>Knowledge Distillation [3]</td>
<td>✗</td>
</tr>
<tr>
<td>Learning under privileged info [4]</td>
<td>✗</td>
</tr>
<tr>
<td>Gaussian Process [5]</td>
<td>✗</td>
</tr>
<tr>
<td>Memory-based CL [6]</td>
<td>✗</td>
</tr>
<tr>
<td>K-priors</td>
<td>✓</td>
</tr>
</tbody>
</table>

Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

<table>
<thead>
<tr>
<th>Method</th>
<th>Wide</th>
<th>Accurate</th>
<th>Quick</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight priors [1]</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>SVMs [2]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Knowledge Distillation [3]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Learning under privileged info [4]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Gaussian Process [5]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Memory-based CL [6]</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-priors</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

<table>
<thead>
<tr>
<th></th>
<th>Wide</th>
<th>Accurate</th>
<th>Quick</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight priors [1]</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>SVMs [2]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Knowledge Distillation [3]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Learning under privileged info [4]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Gaussian Process [5]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Memory-based CL [6]</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-priors</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>


Require storing all past data
Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

\[
\nabla K(w) = \sum_{i \in D} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in D} \nabla f_w^i r_{w*}^i
\]

Residuals \(f_{w*}^i - y_i\)

Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

$$\nabla K(w) = \sum_{i \in \mathcal{D}} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in \mathcal{D}} \nabla f_w^i r_{w*}^i$$

“Avoid past mistakes of the teacher”.

Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

\[ \nabla K(w) = \sum_{i \in D} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in D} \nabla f_w^i r_w^i \]


Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

\[ \nabla K(w) = \sum_{i \in \mathcal{D}} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in \mathcal{D}} \nabla f_w^i r_{w*}^i \]


Results

K-priors need < 2% of past data to match “batch”.

The results are on USPS binary classification with Neural nets. “Replay” uses the same memory but with true outputs.
Results

K-priors only need about 2-5% of the past data to match retraining on full batch.

The results are on USPS binary classification with Neural nets.
Future Directions
Future Directions

• The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
Future Directions

• The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
• This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
Future Directions

• The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
• This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
  – K-prior is a first-order approx. (Gaussian with unknown mean)
Future Directions

• The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
• This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
  – K-prior is a first-order approx. (Gaussian with unknown mean)
  – Extend with posteriors with higher-order sufficient statistics (Gaussian with unknown covariance)
Future Directions

• The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
• This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
  – K-prior is a first-order approx. (Gaussian with unknown mean)
  – Extend with posteriors with higher-order sufficient statistics (Gaussian with unknown covariance)
Future Directions
Future Directions

• Another challenge is what to store and how much memory to allocate
  – Inherent trade-off between speed and accuracy
  – We “have” to reasonable assumptions about the future
  – The “dual space” of the “divergence” plays a key role
Future Directions

• Another challenge is what to store and how much memory to allocate
  – Inherent trade-off between speed and accuracy
  – We “have” to reasonable assumptions about the future
  – The “dual space” of the “divergence” plays a key role

• We are developing “dual representations” are used for Knowledge representation, transfer, and collection
  – A new paper on “memorable past” coming soon
Approximate Bayesian Inference Team

https://team-approx-bayes.github.io/