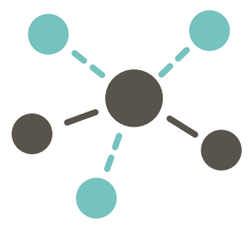
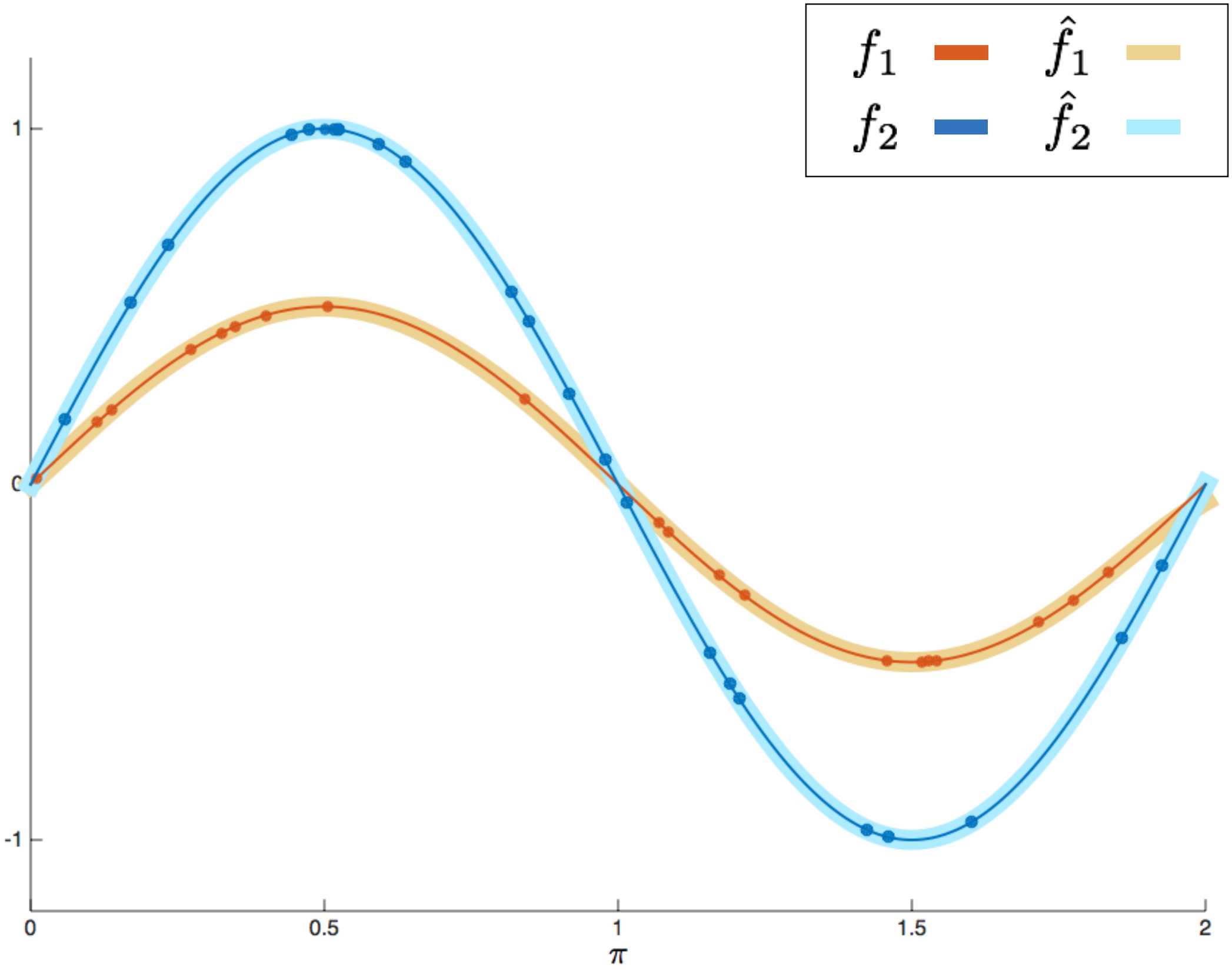
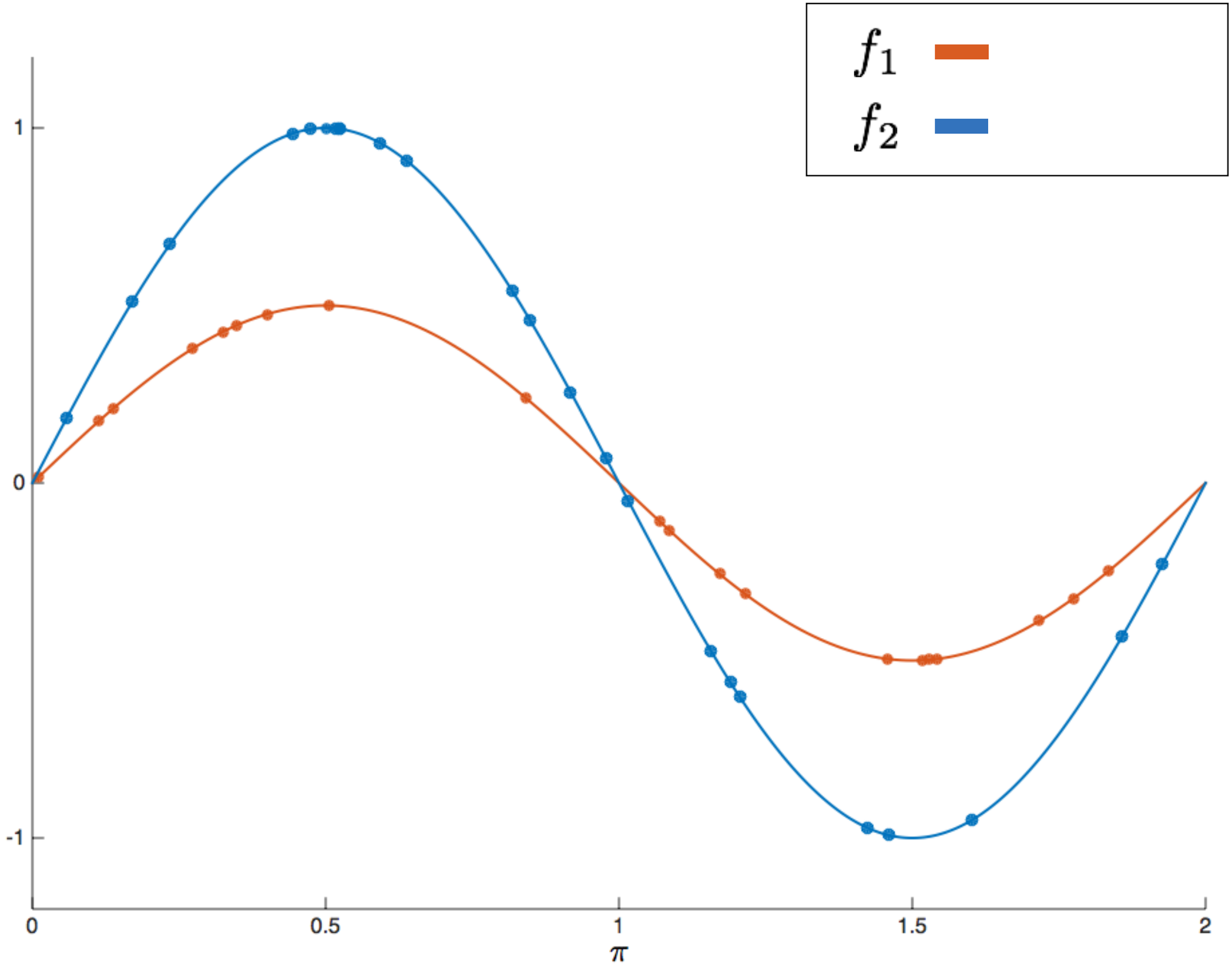
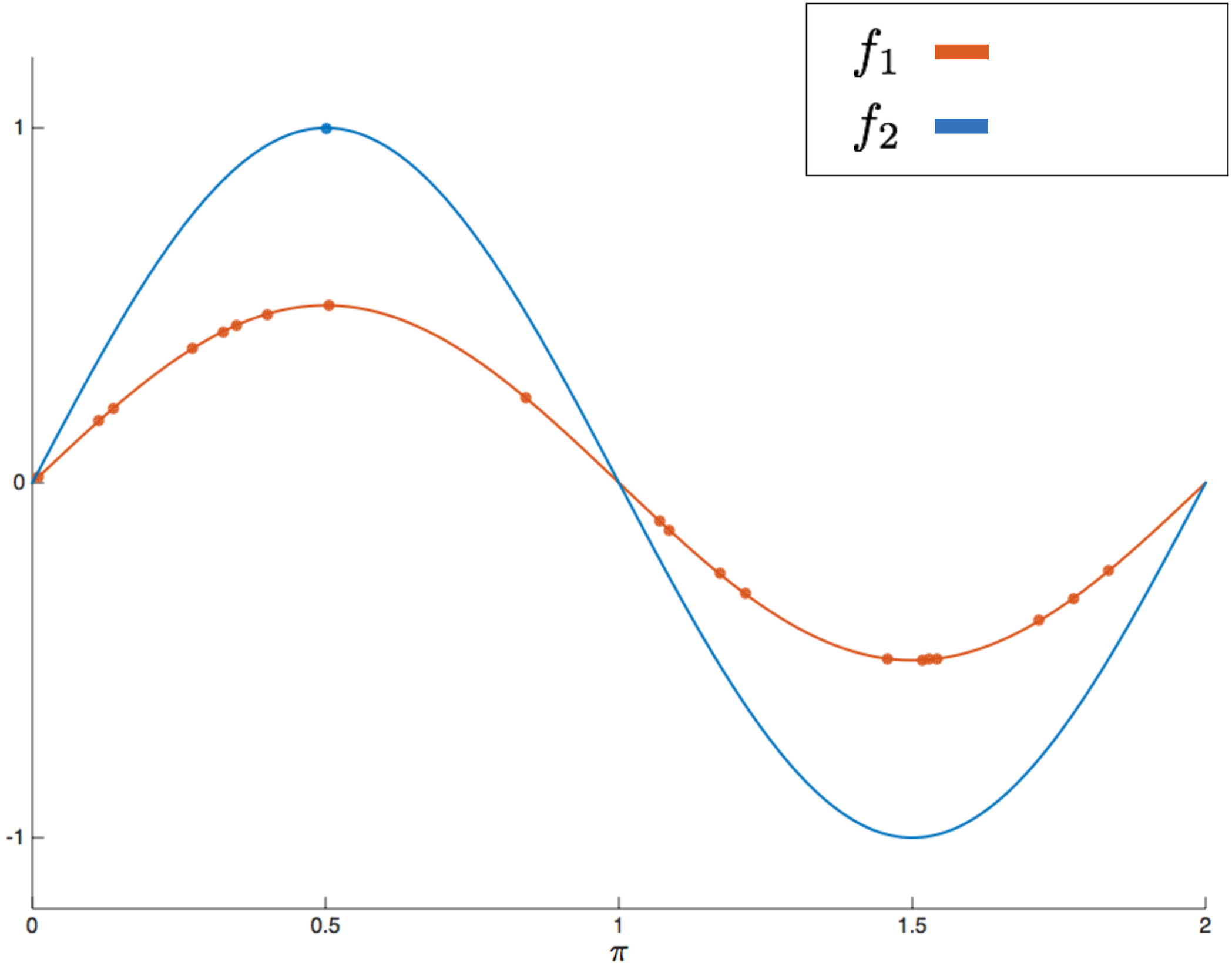
A Unifying Framework for Multi-task Learning

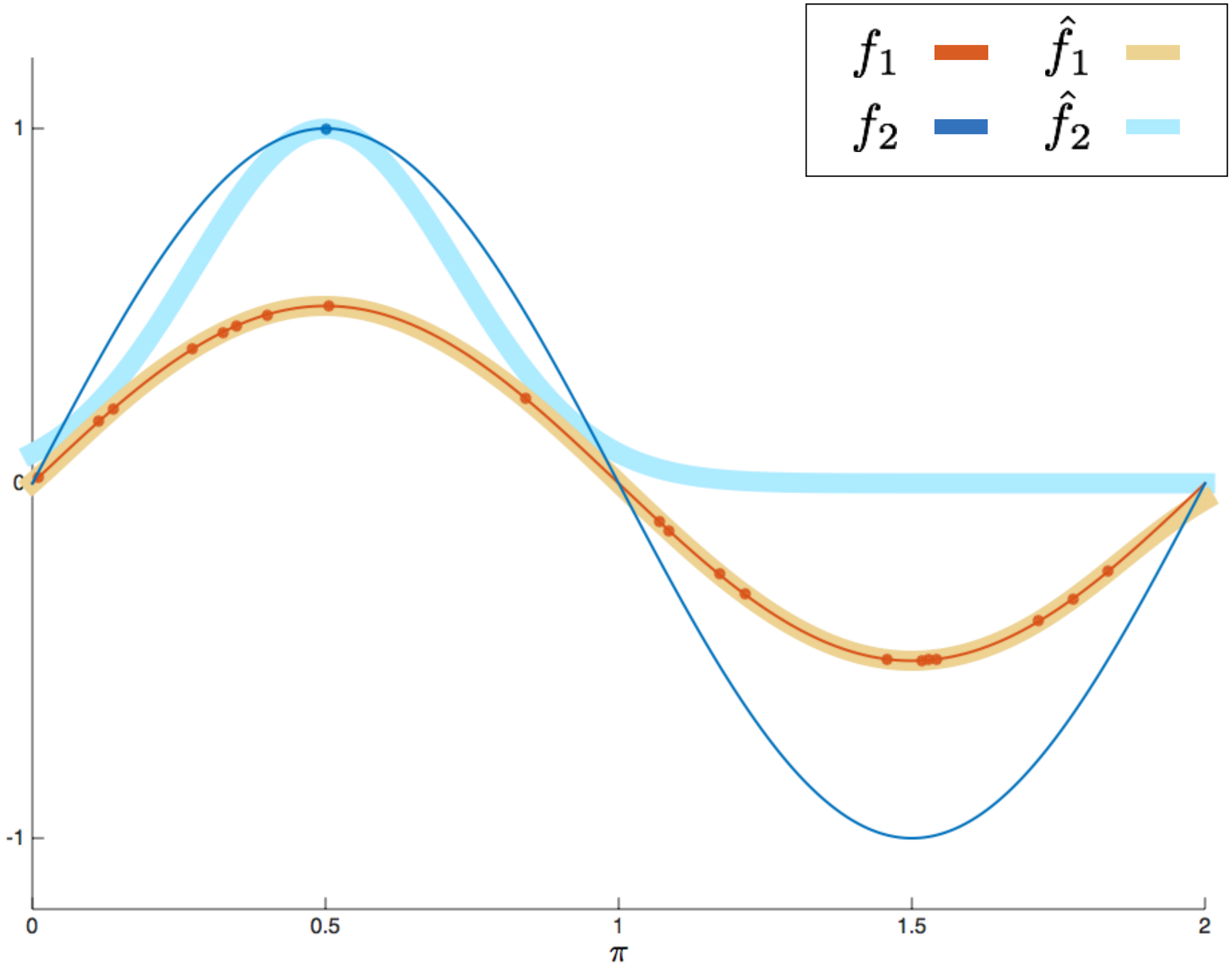


Carlo Ciliberto

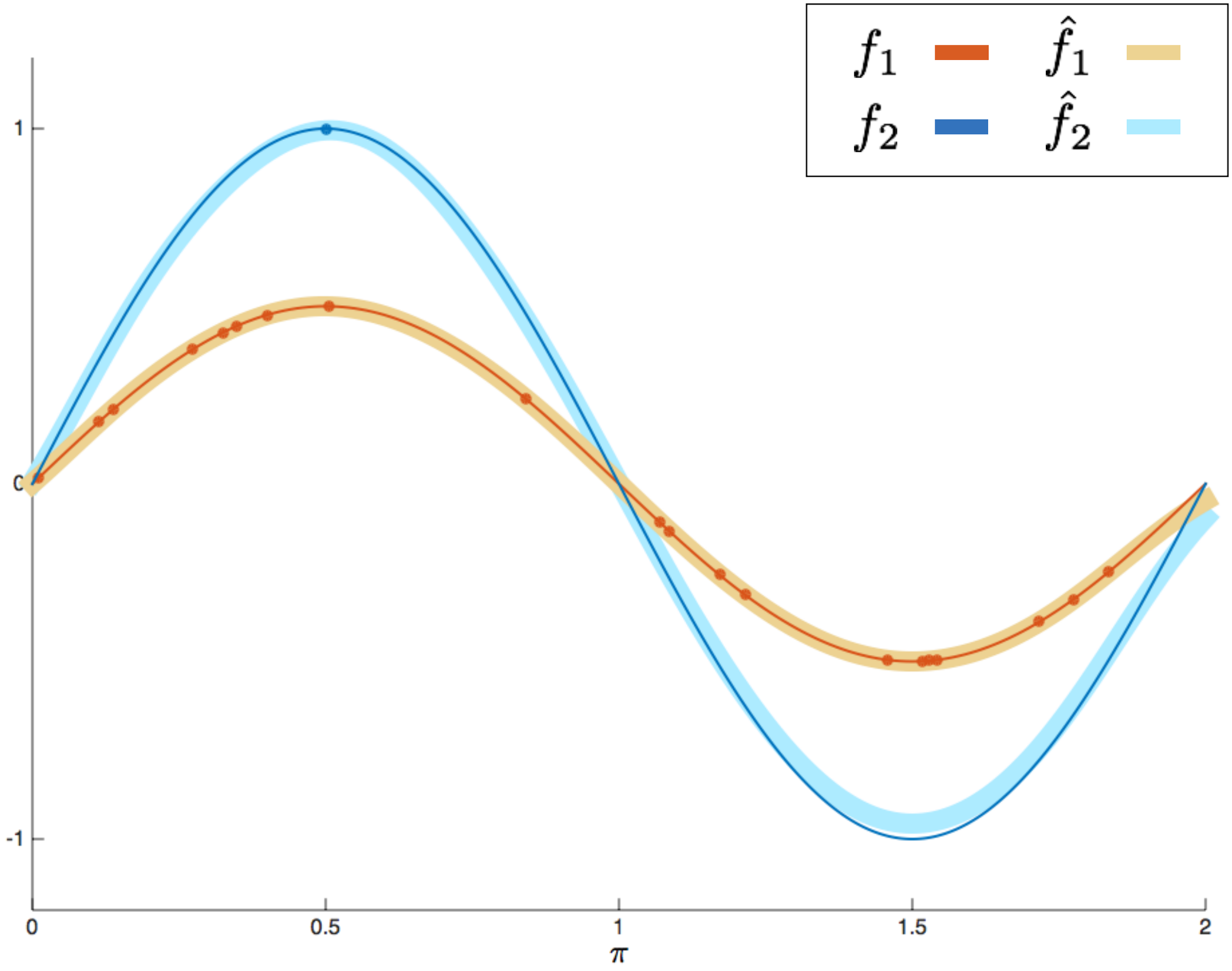

f_1
$\hat{f}_1$
f_2
$\hat{f}_2$
1
0
-1
0
0.5
1
1.5
2
π


f_1
f_2
1
0
-1
0
0.5
1
1.5
2
π


f_1
f_2
1
0
-1
0
0.5
1
1.5
2
π



Without Sharing Information



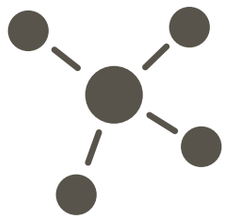
Sharing Information

Multi-task Learning: Assumption

Leveraging on the tasks relations/structure reduces the complexity of the problem

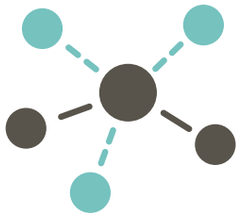
$$
\begin{aligned}
n12 &\to +\infty \\
n17 &\to +\infty \\
n22 &\to +\infty \\
n27 &\to +\infty \\
n32 &\to +\infty \\
n37 &\to +\infty \\
n42 &\to +\infty \\
47 &\to +\infty \\
2 &\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty
\end{aligned}
\qquad
\begin{aligned}
n28 &\to +\infty \\
n33 &\to +\infty \\
n38 &\to +\infty \\
n43 &\to +\infty \\
n48 &\to +\infty \\
n53 &\to +\infty \\
n58 &\to +\infty \\
n63 &\to +\infty \\
n68 &\to +\infty \\
n73 &\to +\infty \\
n78 &\to +\infty \\
n83 &\to +\infty \\
n88 &\to +\infty \\
n93 &\to +\infty \\
n98 &\to +\infty
\end{aligned}
\qquad
\begin{aligned}
n39 &\to +\infty \\
n44 &\to +\infty \\
n49 &\to +\infty \\
n54 &\to +\infty \\
n59 &\to +\infty \\
n64 &\to +\infty \\
n69 &\to +\infty \\
n7 &\to +\infty \\
n &\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty \\
&\to +\infty
\end{aligned}
$$

$$n_{T+1} \to 1$$



Impose known structures

[Evgeniou et al. 2005, Fergus et al. 2010, Kadri et al. 2010, Minh et al 2013, Jayaraman et al., 2014 and many others]



Parametrize and Learn the structure

[Argyriou et al. 2008, Jacob et al. 2009, Zhang et al, 2010 Dinuzzo et al. 2011, Zhong 2012, and many other]

Conjoint Analysis

Transfer Learning

Learning To Learn

Embeddings

Task-relations Learning

Cluster-Task Learning

Structured Output Learning

Multi-Task Learning

Matrix-valued Kernel Learning

Output Covariance Learning

Metric Deformation

Output Representation Learning

Collaborative Filtering

Co-Kriging

To Abstract, Understand & Organize

Can we design a unifying (convex) framework for learning Multiple Tasks and their structure?

Can we design a unifying (convex) framework for learning Multiple Tasks and their structure?

Yes!

Can we provide a general meta-strategy for optimization…

…with convergence guarantees?

Can we provide a general meta-strategy for optimization…

…with convergence guarantees?

Yes!

Can we derive new models of tasks structures from such a framework?

Can we derive new models of tasks structures from such a framework?

Yes!

*Can we design a **unifying** (convex) framework for learning multiple-tasks and their structure?*

*Can we provide a general **meta-strategy** for optimization, with convergence guarantees?*

[Ciliberto et al. - ICML 2015]

*Can we derive **new models** of tasks structures from such a framework?*

[Ciliberto et al. - CVPR 2015]

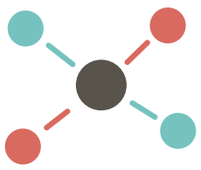
RKHS

for Vector-Valued functions

Examples

A ~ Graph Laplacian

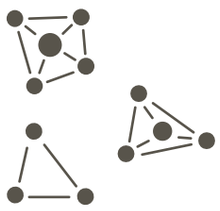
[Evgeniou et al. 2005, Argyriou et al. 2013]



Low dimensional subspace sharing

[Argyriou et al. 2008, Zhang et al. 2010]

$$F(A) = tr(A)$$



Cluster Multi-task learning

[Jacob et al. 2009, kwok et al. 2012]

$$F(A) = \|A\|_c$$

Sparse Kernel Multi-task Learning

[Ciliberto et al. 2015]

$$F(A) = \|A\|_{\ell_1}$$

*Can we design a **unifying** (convex) framework for learning multiple-tasks and their structure?*

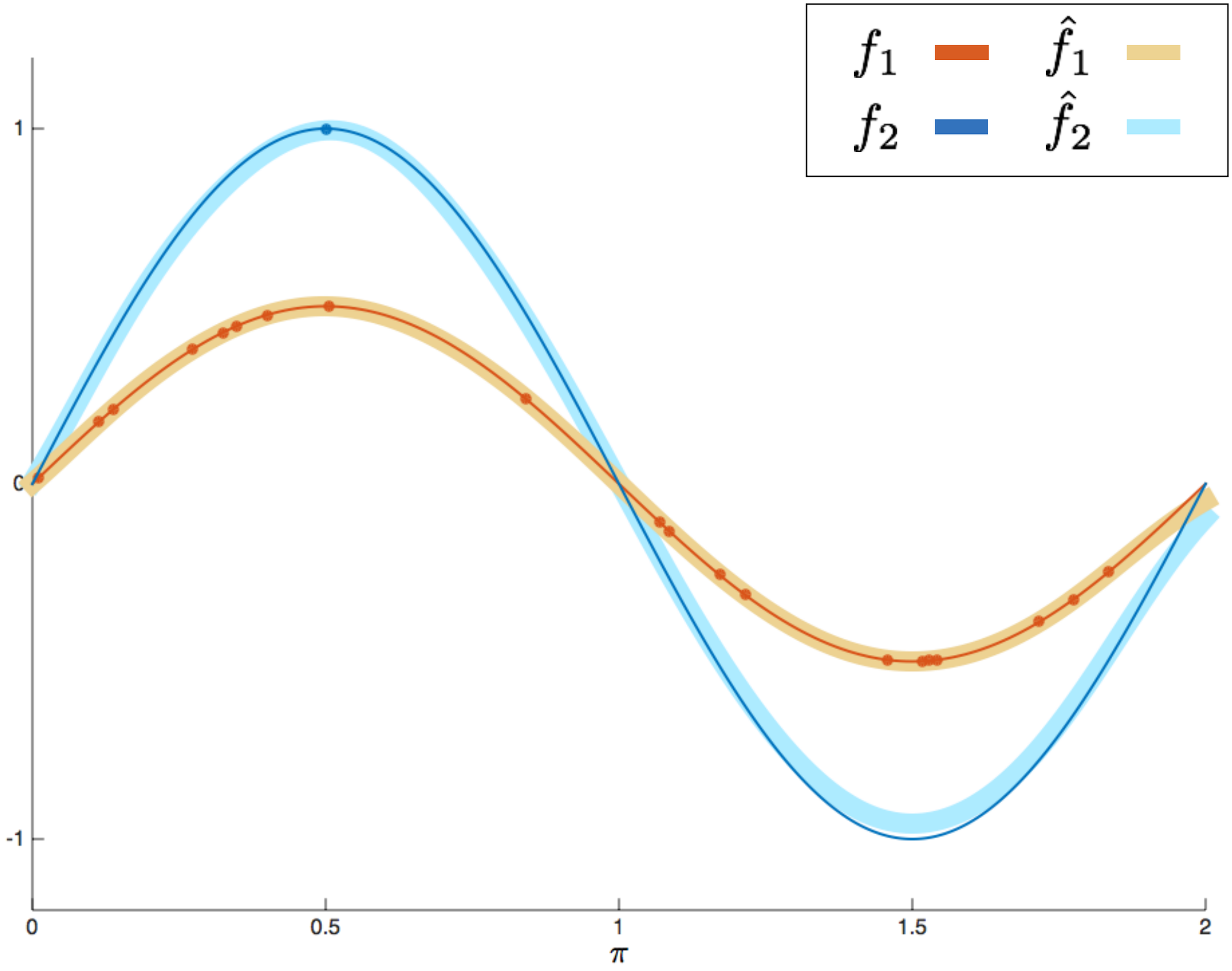
*Can we provide a general **meta-strategy** for optimization, with convergence guarantees?*

[Ciliberto et al. - ICML 2015]

*Can we derive **new models** of tasks structures from such a framework?*

[Ciliberto et al. - CVPR 2015]

Are we done?


f_1
$\hat{f}_1$
f_2
$\hat{f}_2$
1
0
-1
0
0.5
1
1.5
2
π

$$k(x, z) A_{ts}$$

$$\Gamma(x, z)_{ts}$$

Can we find a parametrization for all Operator-valued Kernels?

Can we still learn them?

Can we find a parametrization for all Operator-valued Kernels?

Can we still learn them?

Spoiler alert: Yes!

[Ciliberto et al. - In Preparation]

Take home messages

Take home messages

MTL

Multi Task Learning

If tasks are related, solving them jointly can be much more favorable!

Take home messages

MTL

Multi Task Learning

If tasks are related, solving them jointly can be much more favorable!

$\mathbf{k}(\mathbf{x}, \mathbf{z})\mathbf{A}$

RKHS for vector-valued functions

Are the way to go! you can:

Take home messages

MTL

Multi Task Learning

If tasks are related, solving them jointly can be much more favorable!

$\mathbf{k}(\mathbf{x}, \mathbf{z})\mathbf{A}$

RKHS for vector-valued functions

Are the way to go! you can:

Impose prior knowledge on the structure

By designing a suitable structure matrix A

Learn the relations!

Imposing a structure penalty F(A) on the problem

Take home messages

MTL

Multi Task Learning
If tasks are related, solving them jointly can be much more favorable!

$\mathbf{k}(\mathbf{x}, \mathbf{z})\mathbf{A}$

RKHS for vector-valued functions
Are the way to go! you can:

Impose prior knowledge on the structure
By designing a suitable structure matrix A

Learn the relations!
Imposing a structure penalty F(A) on the problem

Future Work

More complex intra-task relations
Impose or learn more complex input-output relations