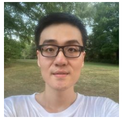
Theseus

A library for differentiable nonlinear optimization


Luis Pineda


Taosha Fan


Maurizio Monge


Shobha Venkataraman


Paloma Sodhi

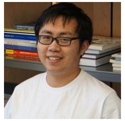

Ricky T. Q. Chen


Joseph Ortiz


Daniel DeTone


Austin Wang


Stuart Anderson


Jing Dong


Brandon Amos


Mustafa Mukadam

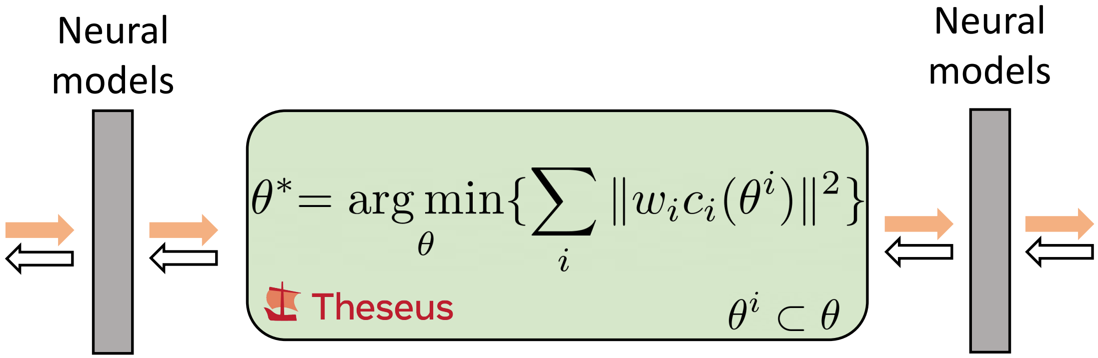

```
pip install theseus-ai
```

Meta

NEURAL INFORMATION PROCESSING SYSTEMS

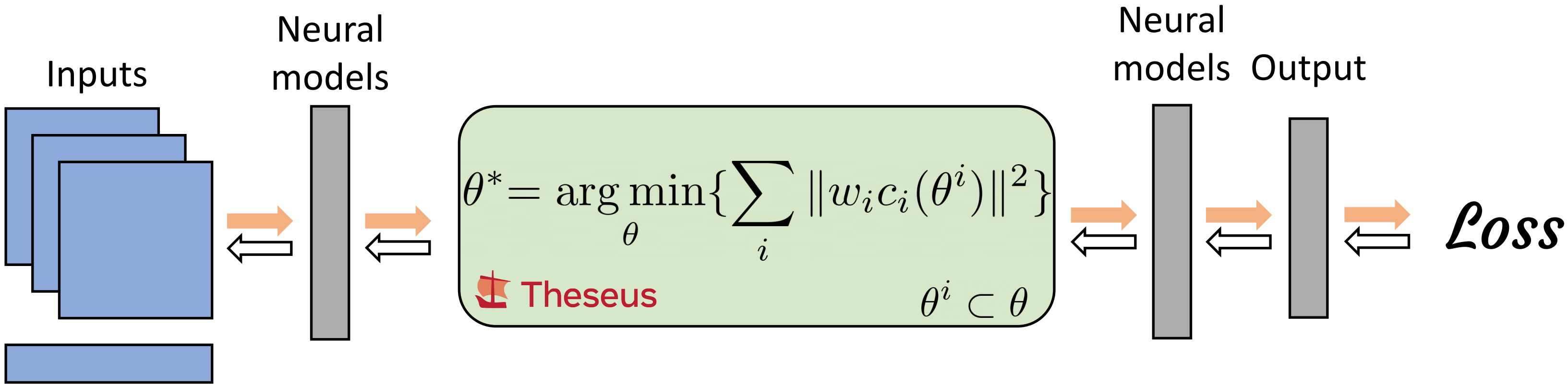
https://sites.google.com/view/theseus-ai

What is Theseus?



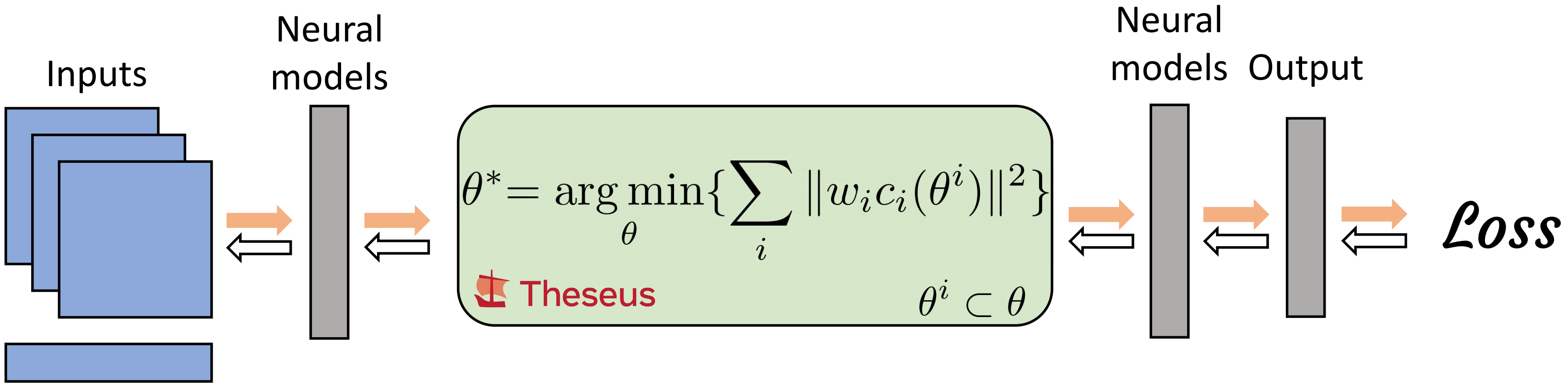
Theseus is a library for building custom nonlinear optimization layers in PyTorch

What is Theseus?



Theseus is a library for building custom nonlinear optimization layers in PyTorch to support constructing end-to-end differentiable architectures

What is Theseus?



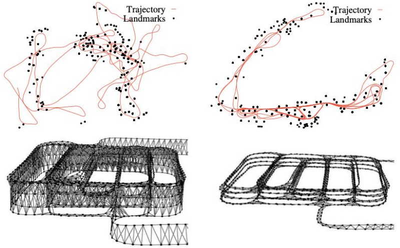
Theseus is an efficient application-agnostic library for building custom nonlinear optimization layers in PyTorch to support constructing various problems in robotics and vision as end-to-end differentiable architectures

Structure-from-Motion Revisited

Johannes L. Schönberger[1,2*], Jan-Michael Frahm[1]

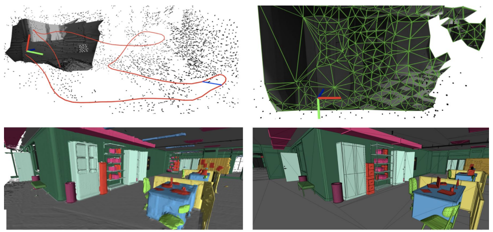
g²o: A General Framework for Graph Optimization

Rainer Kümmerle Giorgio Grisetti Hauke Strasdat Kurt Konolige Wolfram Burgard



Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping

Antoni Rosinol, Marcus Abate, Yun Chang, Luca Carlone



Tracking many objects with many sensors

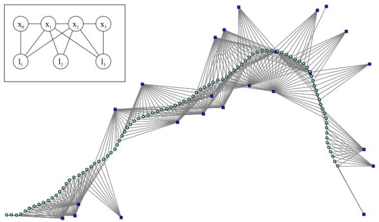
Hanna Pasula and Stuart Russell Michael Ostland and Ya'acov Ritov*

Generalized-ICP

Aleksandr V. Segal Dirk Haehnel Sebastian Thrun

Square Root SAM
Simultaneous Localization and Mapping via Square Root Information Smoothing

Frank Dellaert and Michael Kaess

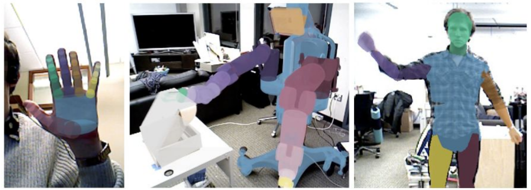


A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control

Markus Giftthaler[1], Michael Neunert[1], Markus Stäuble[1], Jonas Buchli[1] and Moritz Diehl[2]

DART: Dense Articulated Real-Time Tracking

Tanner Schmidt, Richard Newcombe, Dieter Fox

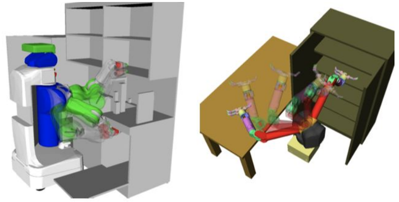


Recovering 3D Shape and Motion from Image Streams using Non-Linear Least Squares

Richard Szeliski and Sing Bing Kang

Continuous-time Gaussian process motion planning via probabilistic inference

Mustafa Mukadam*, Jing Dong*, Xinyan Yan, Frank Dellaert and Byron Boots

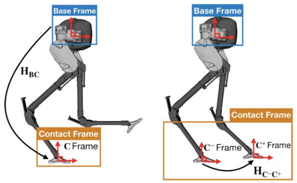


Bundle Adjustment — A Modern Synthesis

Bill Triggs[1], Philip McLauchlan[2], Richard Hartley[3] and Andrew Fitzgibbon[4]

Hybrid Contact Preintegration for Visual-Inertial-Contact State Estimation Using Factor Graphs

Ross Hartley, Maani Ghaffari Jadidi, Lu Gan, Jiunn-Kai Huang, Jessy W. Grizzle, and Ryan M. Eustice

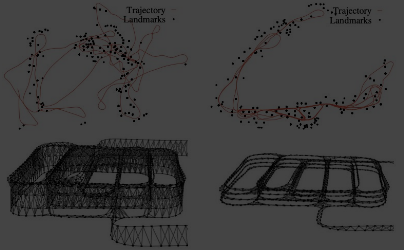


Structure-from-Motion Revisited

Johannes L. Schönberger[1,2]*, Jan-Michael Frahm[1]

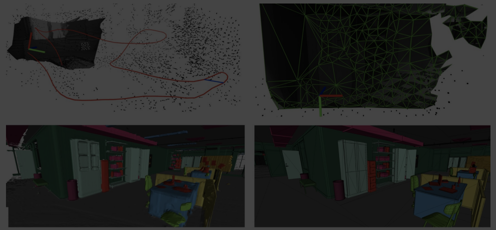
g²o: A General Framework for Graph Opti

Rainer Kümmerle Giorgio Grisetti Hauke Strasdat Kurt Konolige



Kimera: an Open-Source Library for Real-T
Metric-Semantic Localization and Mappin

Antoni Rosinol, Marcus Abate, Yun Chang, Luca Carlone



Tracking many objects with many sensors

Hanna Pasula and Stuart Russell Michael Ostland and Ya'acov Ritov*

Generalized-ICP

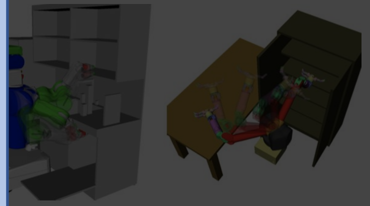
Aleksandr V. Segal Dirk Haehnel Sebastian Thrun

Recovering 3D Shape and Motion from Image Streams using Non-Linear Least Squares

Richard Szeliski and Sing Bing Kang

uous-time Gaussian process
nning via probabilistic

', Jing Dong*, Xinyan Yan, Frank Dellaert and Byron Boots

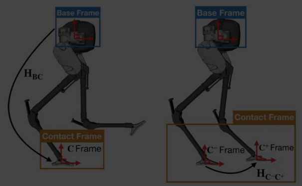


Adjustment — A Modern Synthesis

cLauchlan[2], Richard Hartley[3] and Andrew Fitzgibbon[4]

act Preintegration for Visual-Inertial-Contact State
Estimation Using Factor Graphs

ani Ghaffari Jadidi, Lu Gan, Jiunn-Kai Huang, Jessy W. Grizzle, and Ryan M. Eustice



SLAM

Bundle adjustment

Structure from motion

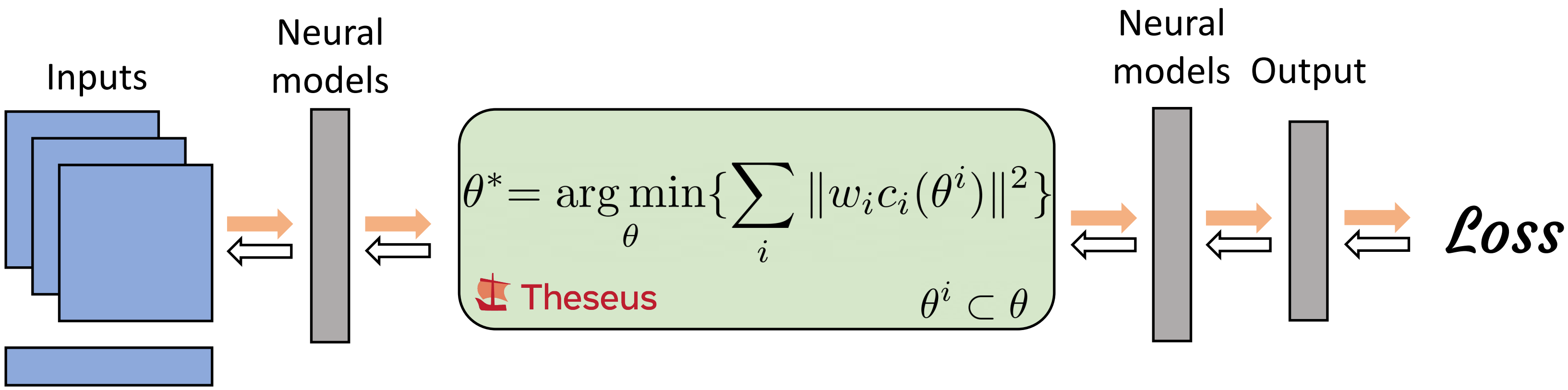
Tracking and estimation

Motion planning

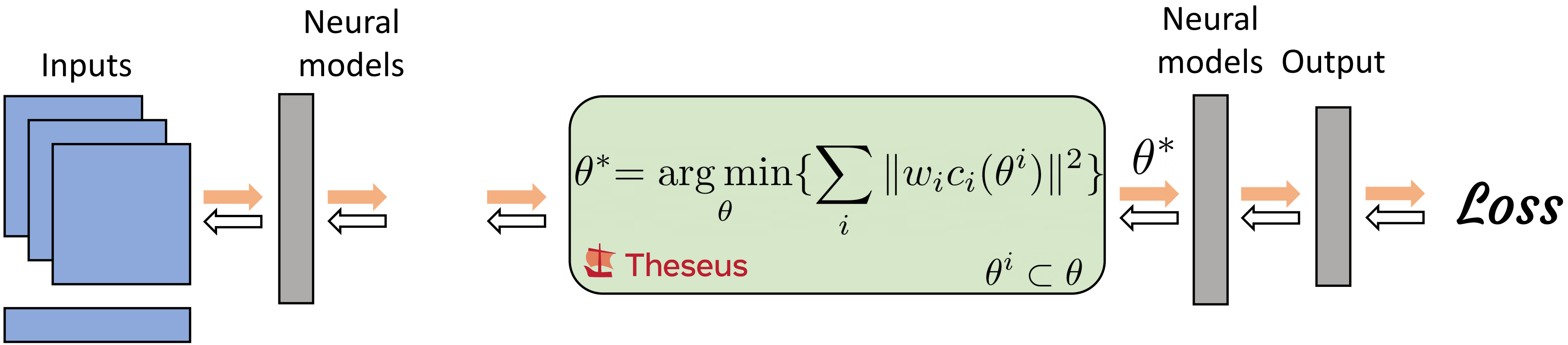
Optimal control

…

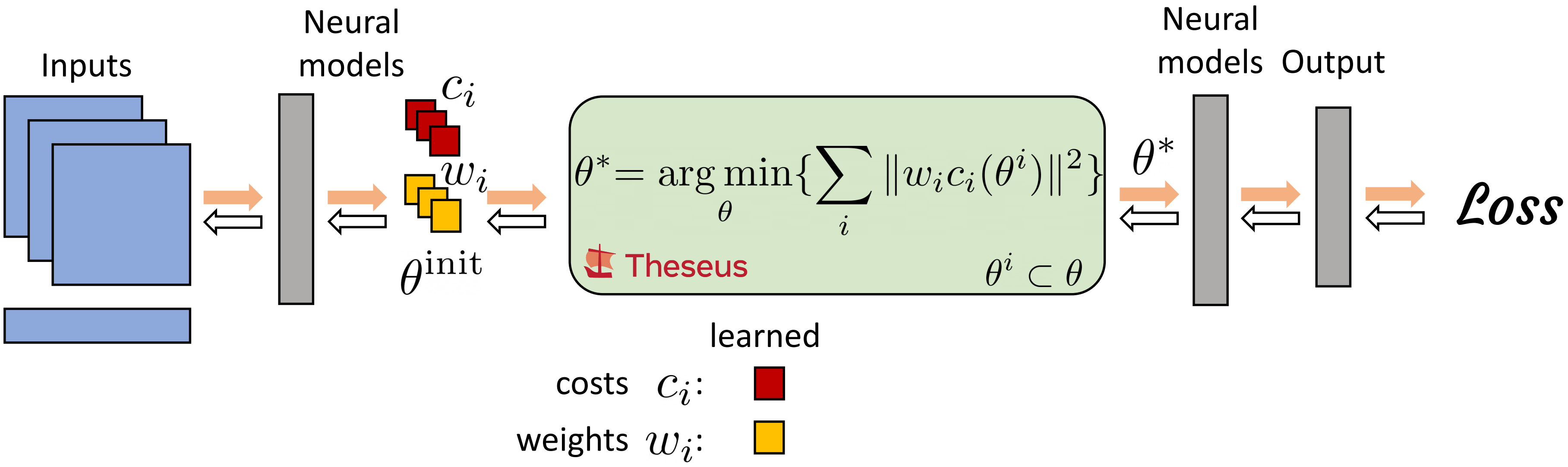
What is Theseus?



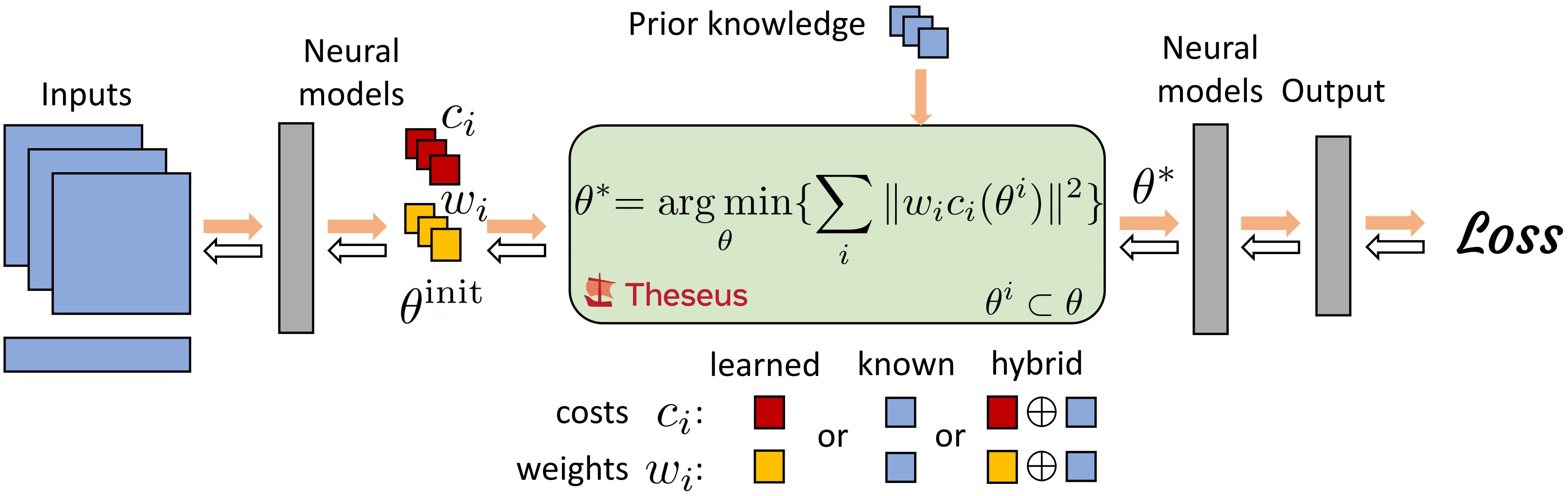
What is Theseus?



What is Theseus?

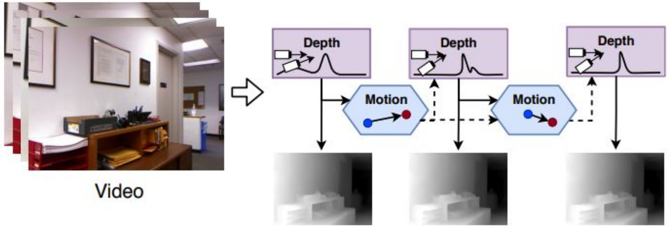


What is Theseus?



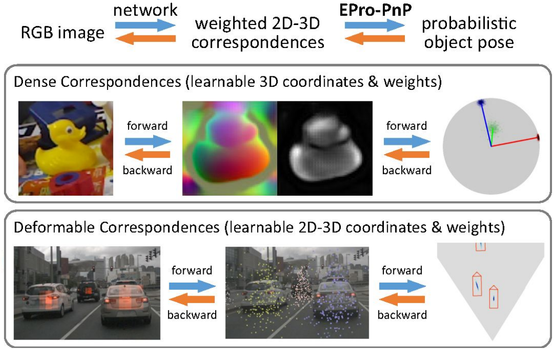
Taking a Deeper Look at the Inverse Compositional Algorithm

Zhaoyang Lv[1,2] Frank Dellaert[1] James M. Rehg[1] Andreas Geiger[2]



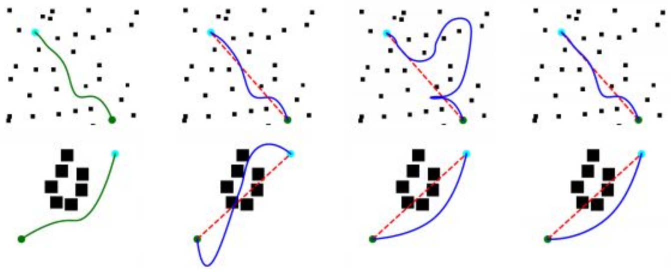
DeepV2D: Video to Depth with Differentiable Structure from Motion

Zachary Teed Jia Deng



EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation

Hansheng Chen[1,2,*] Pichao Wang[2,†] Fan Wang[2] Wei Tian[1,†] Lu Xiong[1] Hao Li[2]
[1]School of Automotive Studies, Tongji University [2]Alibaba Group

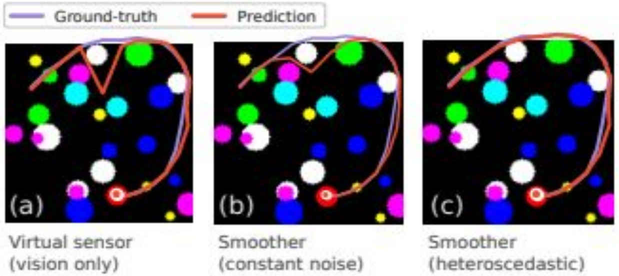


Differentiable Gaussian Process Motion Planning

Mohak Bhardwaj[1], Byron Boots[1], and Mustafa Mukadam[2]

∇SLAM: Automagically differentiable SLAM
https://gradslam.github.io

Krishna Murthy J.[*1,2,3], Soroush Saryazdi[*4], Ganesh Iyer[5], and Liam Paull[†1,2,3,6]

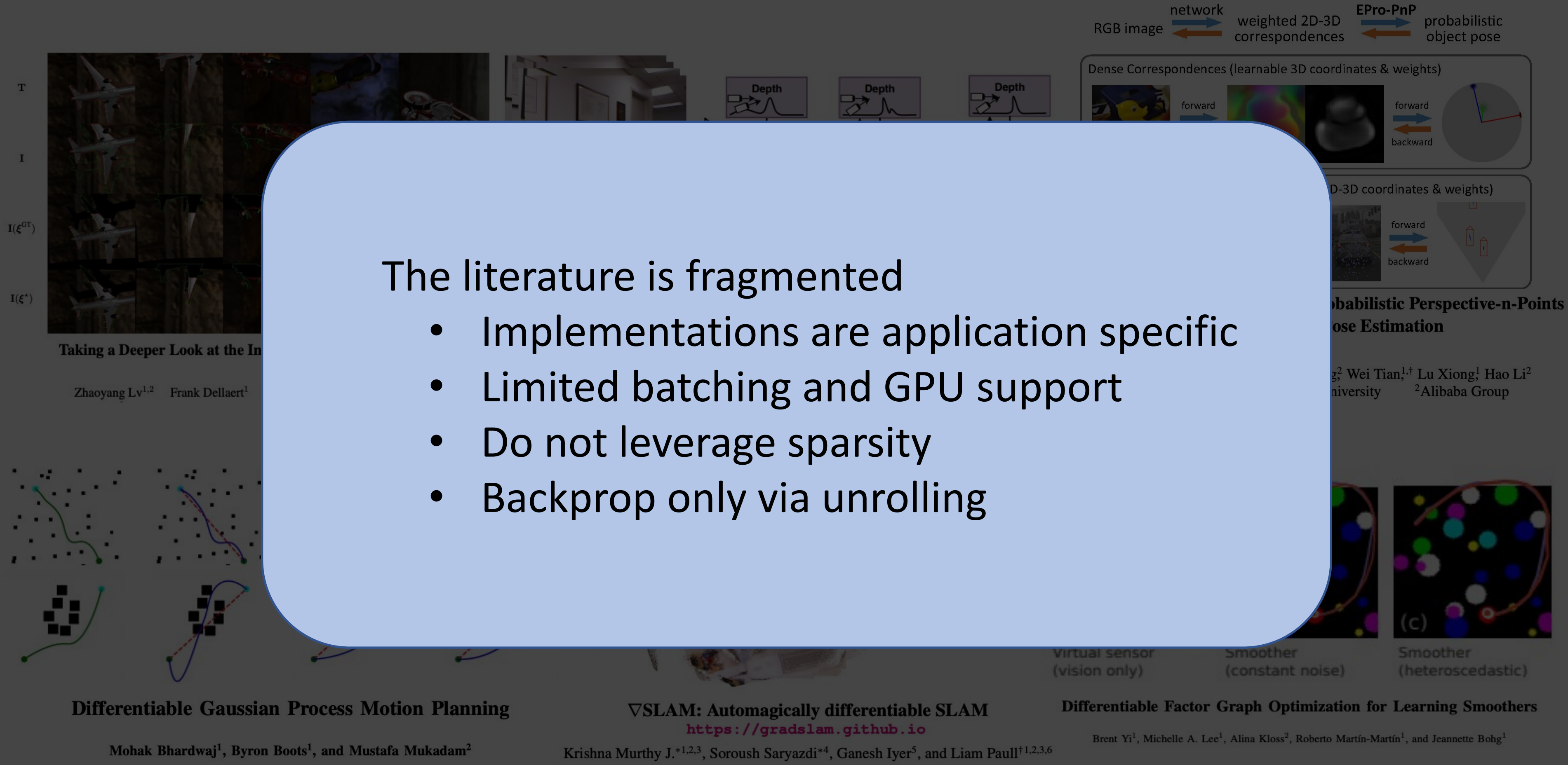


Differentiable Factor Graph Optimization for Learning Smoothers

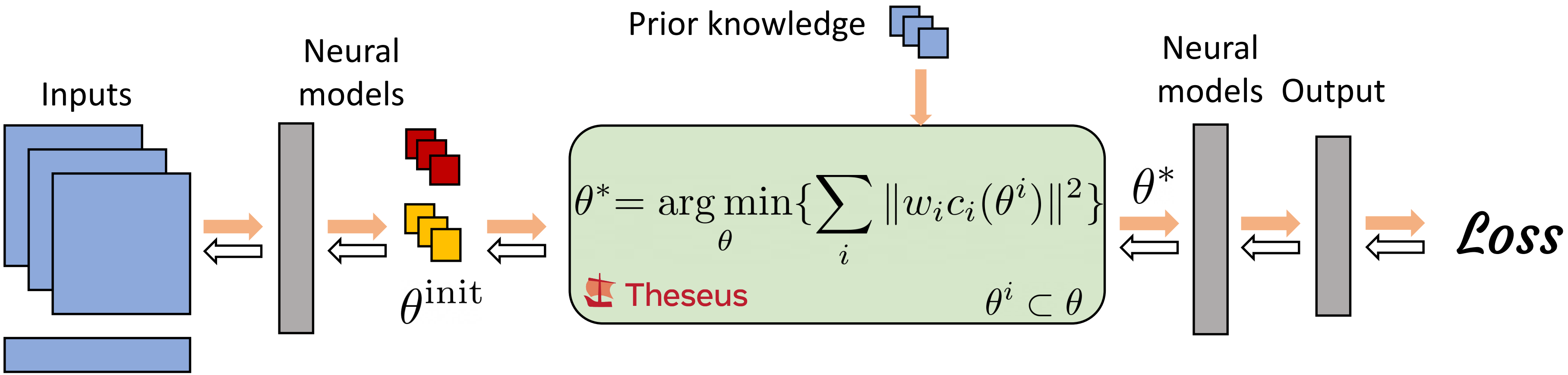
Brent Yi[1], Michelle A. Lee[1], Alina Kloss[2], Roberto Martín-Martín[1], and Jeannette Bohg[1]

The literature is fragmented

- Implementations are application specific
- Limited batching and GPU support
- Do not leverage sparsity
- Backprop only via unrolling



What is Theseus?

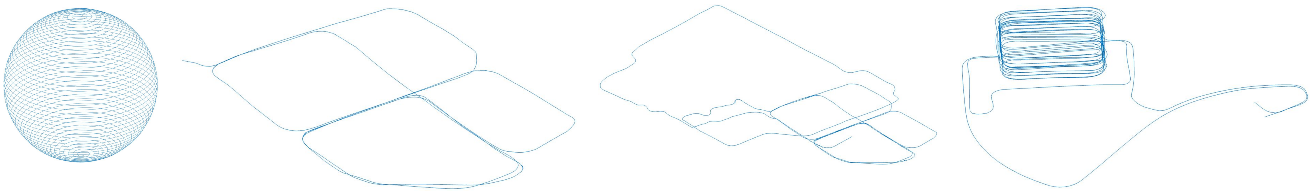


Theseus is an efficient application-agnostic library for building custom nonlinear optimization layers in PyTorch to support constructing various problems in robotics and vision as end-to-end differentiable architectures

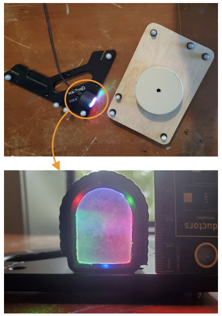
Application Agnostic	**Second-Order Nonlinear Optimizers**	**Lie Groups**	**Cost Functions**
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	**Sparse Linear Solvers**	**Parallelization**	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

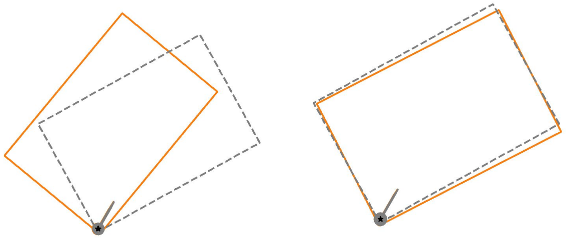
Application Agnostic	**Second-Order Nonlinear Optimizers**	**Lie Groups**	**Cost Functions**
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	**Sparse Linear Solvers**	**Parallelization**	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

Pose graph optimization

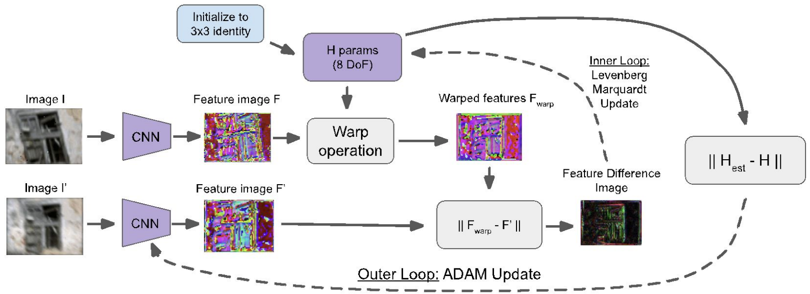


Tactile state estimation

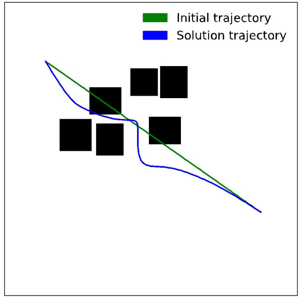


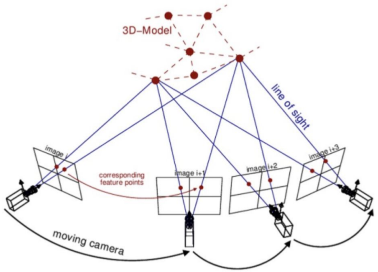


Homography estimation



Motion Planning

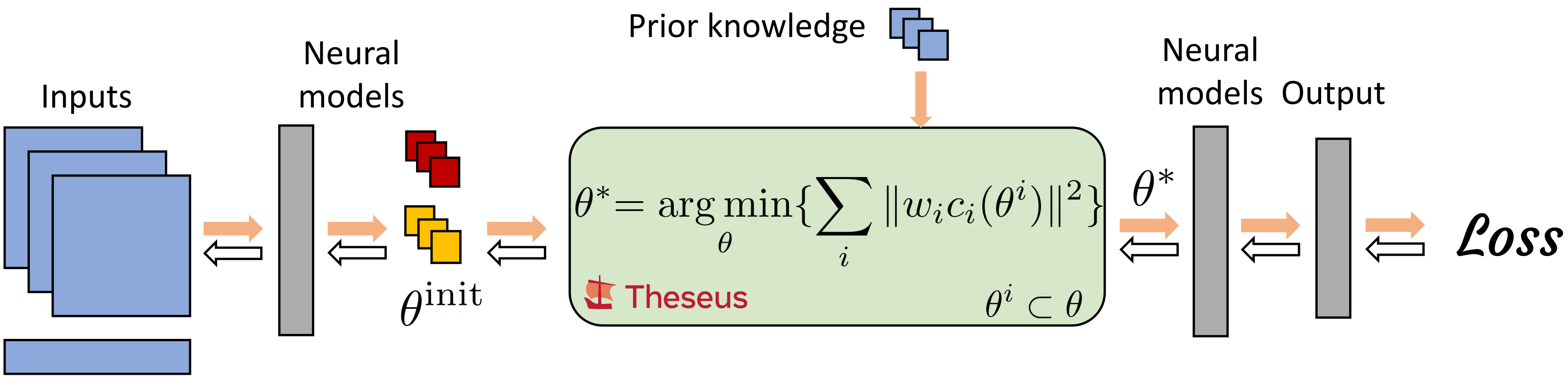




Bundle adjustment

Application Agnostic	**Second-Order Nonlinear Optimizers**	**Lie Groups**	**Cost Functions**
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	**Sparse Linear Solvers**	**Parallelization**	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

Application Agnostic	**Second-Order Nonlinear Optimizers**	**Lie Groups**	**Cost Functions**
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	**Sparse Linear Solvers**	**Parallelization**	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

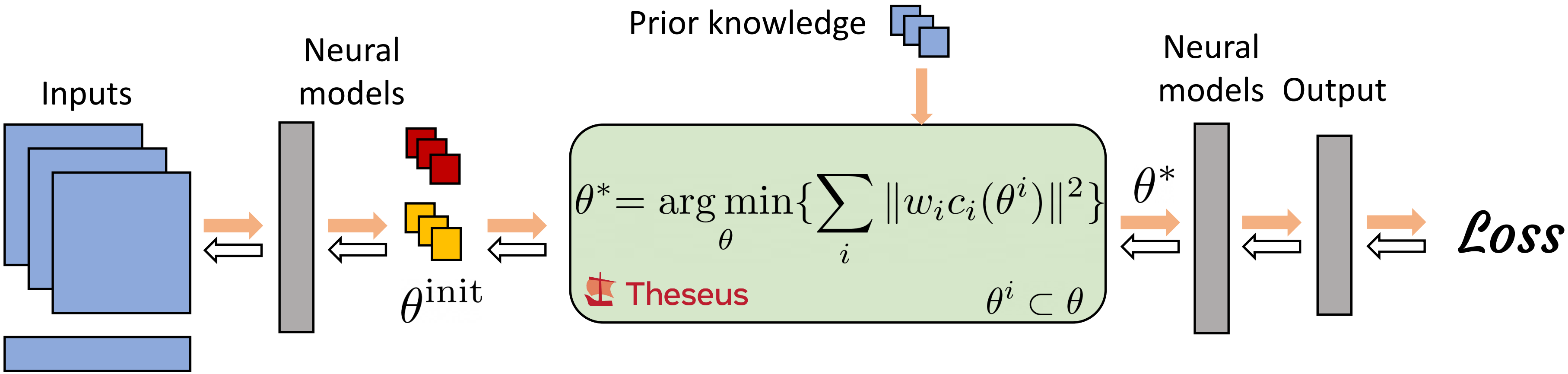


Step 1: Linearize $A\delta\theta = b$

Step 2: Linear solve $\delta\theta$

Step 3: Update $\theta \leftarrow \theta + \delta\theta$

repeat

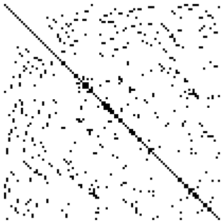


Step 1: Linearize $A\delta\theta = b$

Step 2: Linear solve $\delta\theta$

Step 3: Update $\theta \leftarrow \theta + \delta\theta$

repeat

$A =$ 

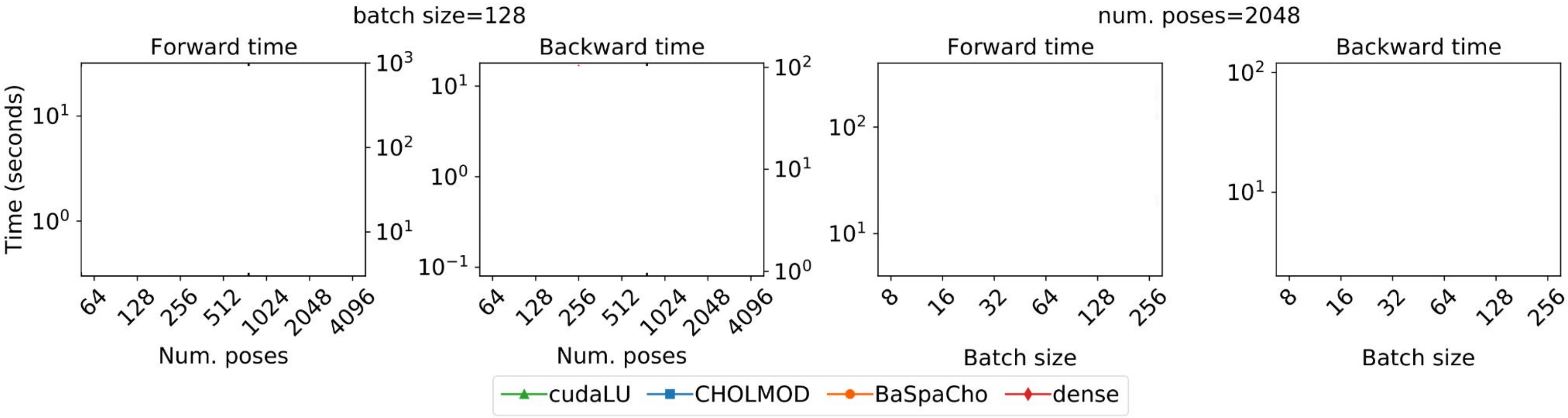
torch.solve() treats *A* as a dense matrix

Sparse vs Dense solvers

Pose Graph Optimization

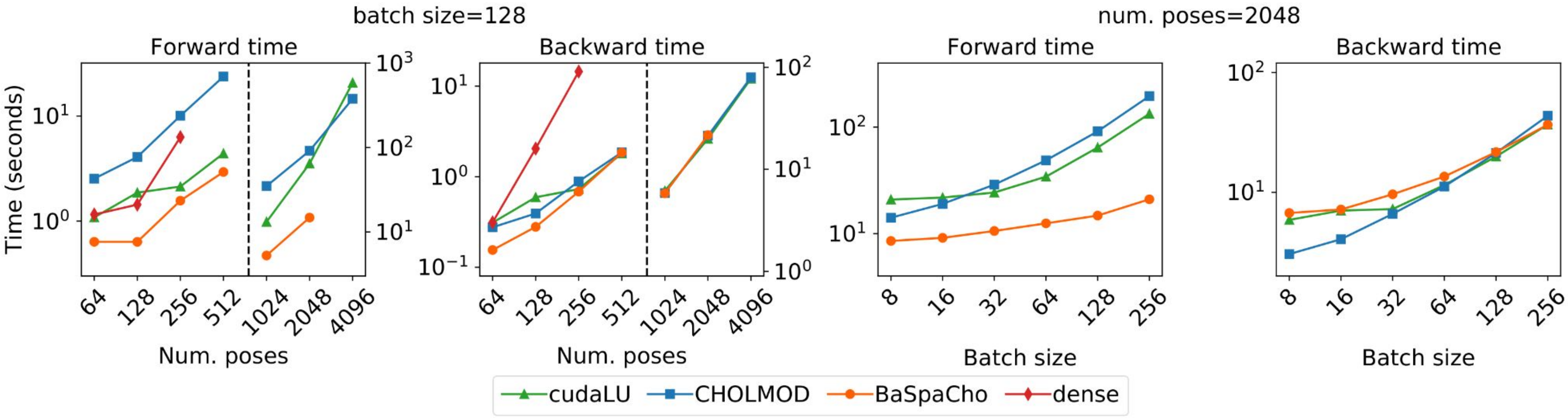
Sparse vs Dense solvers

Pose Graph Optimization



Sparse vs Dense solvers

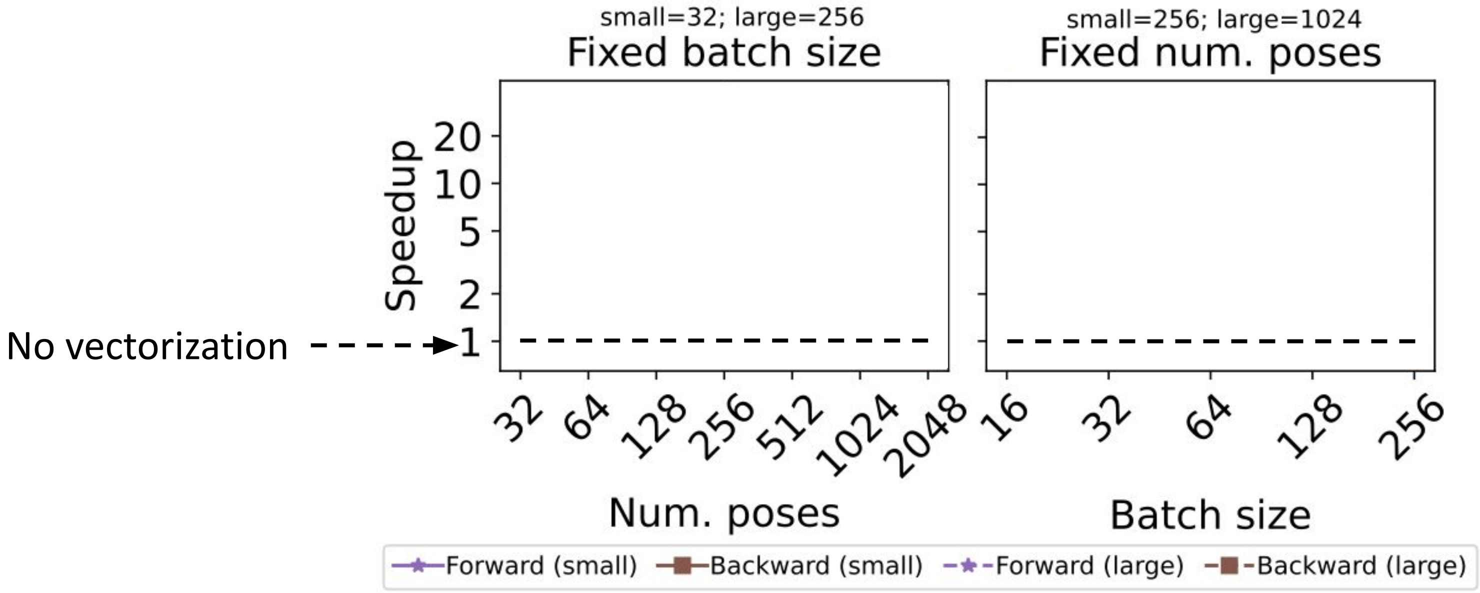
Pose Graph Optimization



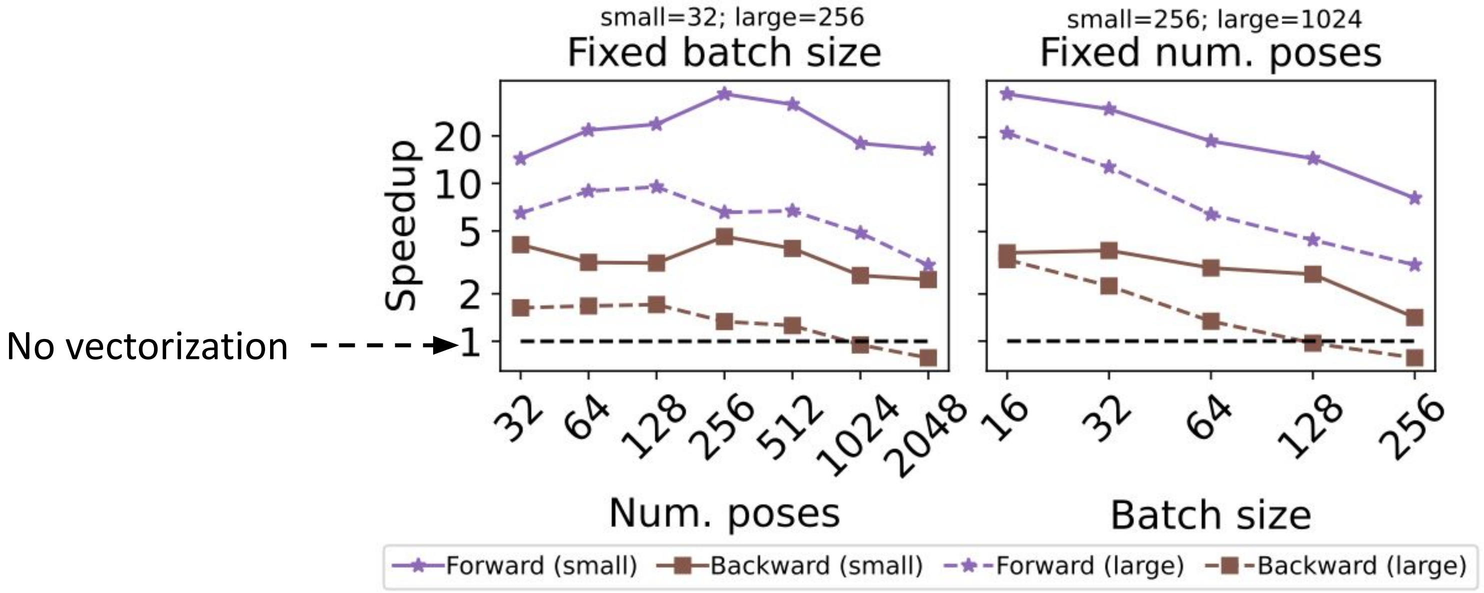
- Scales to 256 batch x 4096 poses on a standard GPU
- Dense solvers are slow and run out of memory

Application Agnostic	Second-Order Nonlinear Optimizers	Lie Groups	Cost Functions
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	Sparse Linear Solvers	**Parallelization**	Backward Modes
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

Automatic vectorization



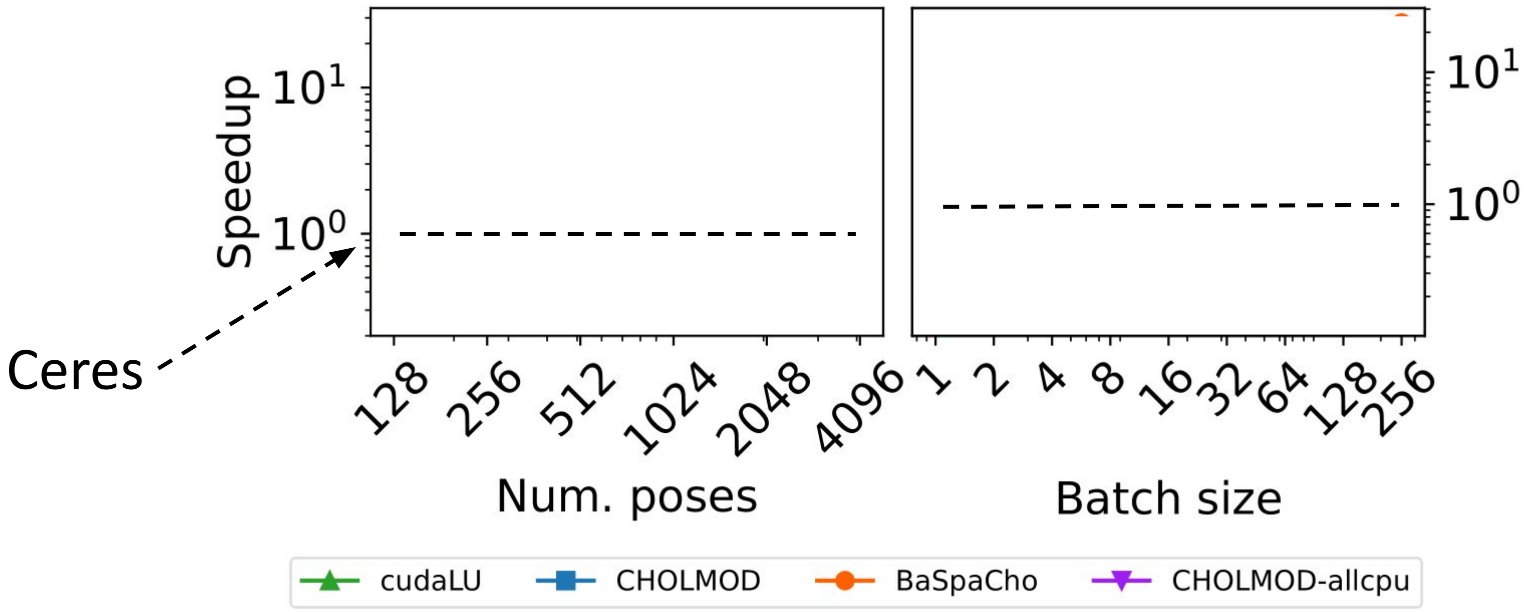
Automatic vectorization



- Up to an order of magnitude speed-up

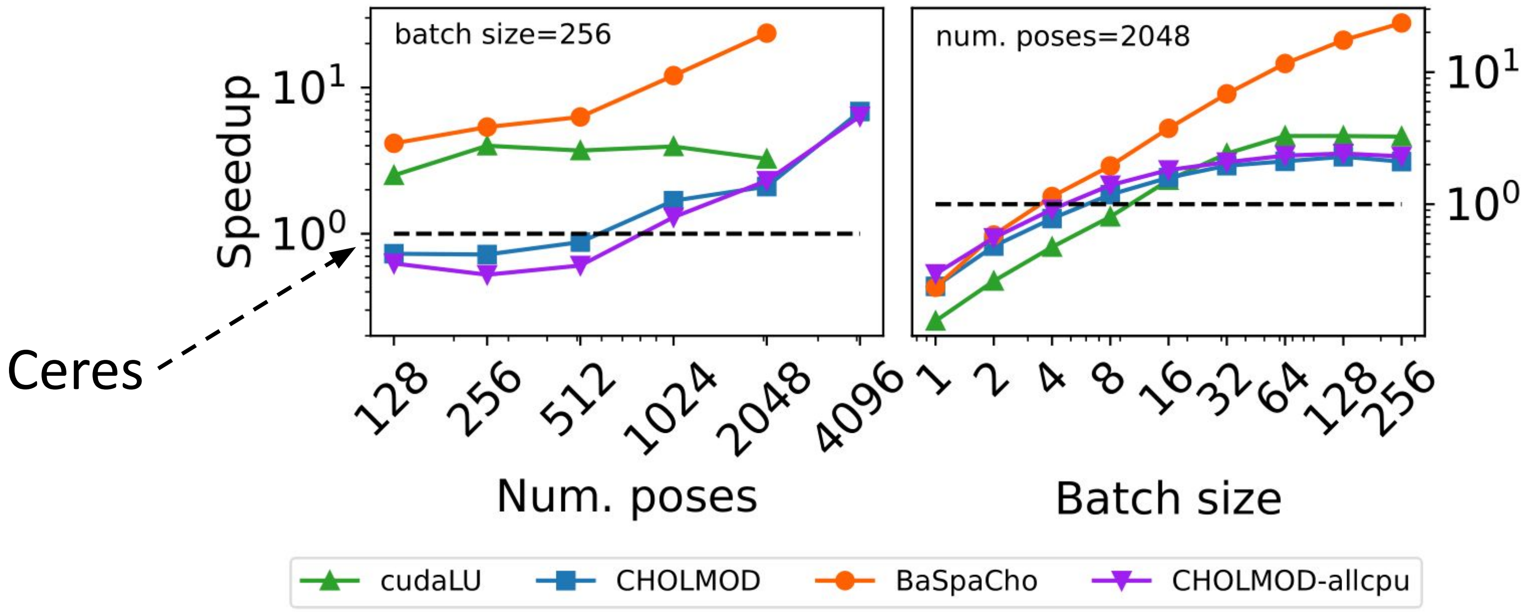
Theseus forward vs Ceres

Pose Graph Optimization



Theseus forward vs Ceres

Pose Graph Optimization

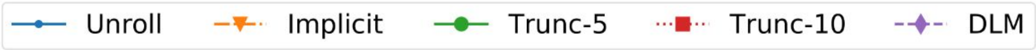


- Up to 20x speed-up over Ceres

Application Agnostic	Second-Order Nonlinear Optimizers	Lie Groups	Cost Functions
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	Sparse Linear Solvers	Parallelization	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

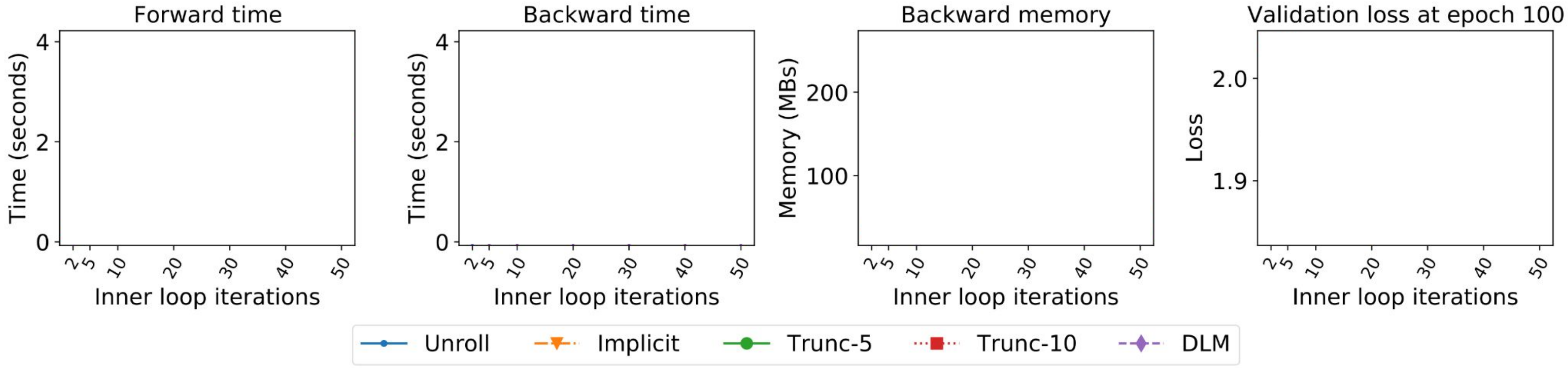
Backward modes

Tactile state estimation



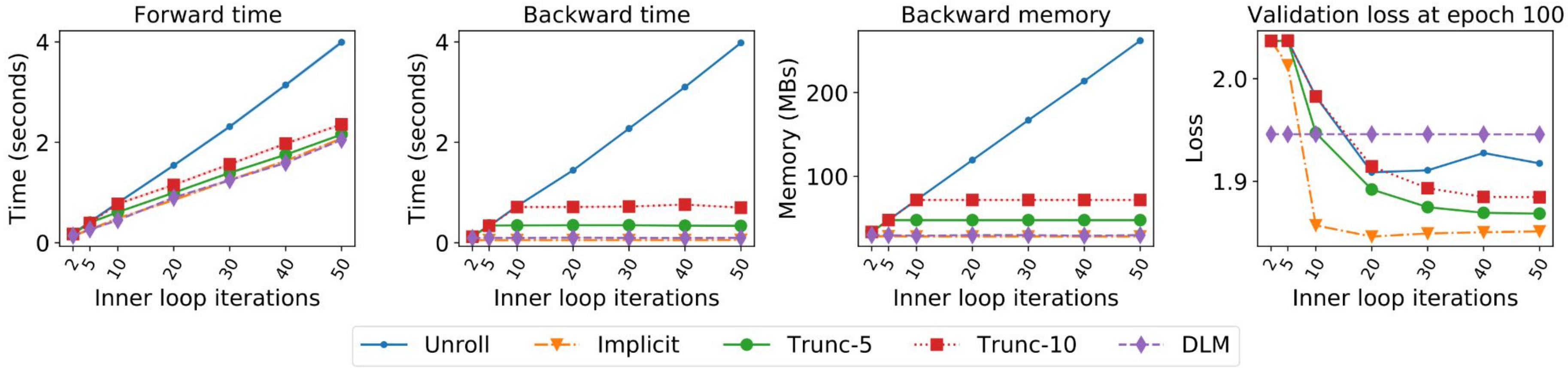
Backward modes

Tactile state estimation



Backward modes

Tactile state estimation

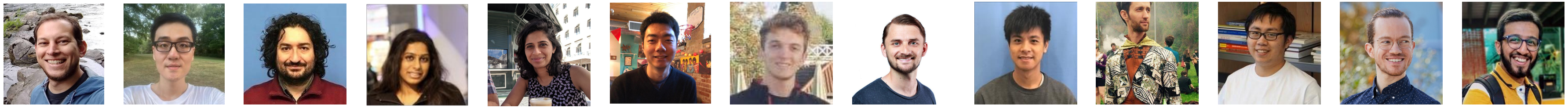


- Implicit needs only constant time and memory
- Implicit gives better gradients
  - Even with few inner loop iterations

Application Agnostic	**Second-Order Nonlinear Optimizers**	**Lie Groups**	**Cost Functions**
	Gauss-Newton, LM	SO2, SE2, SO3, SE3	Measurements, Collision, Kinematics, Dynamics
Efficient	**Sparse Linear Solvers**	**Parallelization**	**Backward Modes**
	CHOLMOD, LU, BaSpaCho	Batching, GPU, Auto Vectorization	Implicit, Truncated, Unroll, Direct Loss

Theseus

A library for differentiable nonlinear optimization



Luis Pineda | Taosha Fan | Maurizio Monge | Shobha Venkataraman | Paloma Sodhi | Ricky T. Q. Chen | Joseph Ortiz | Daniel DeTone | Austin Wang | Stuart Anderson | Jing Dong | Brandon Amos | Mustafa Mukadam

```
pip install theseus-ai
```

Meta

NEURAL INFORMATION PROCESSING SYSTEMS

https://sites.google.com/view/theseus-ai