PhD Dissertation Defense
6th April 2021

POLITECNICO
MILANO 1863

On Iterative and Conditional Computation for Visual Representation Learning

Marco Ciccone

Advisor: Prof. Matteo Matteucci
Co-Advisor: Dr. Jonathan Masci
Tutor: Prof. Cristina Silvano

Politecnico di Milano
Dipartimento di Elettronica, Informatica e Bioingegneria
Doctoral Programme in Information Technology - Cycle XXXII

Machine Learning

“Learning from experience how to perform a given task

that has to be automatized by a machine.”

Supervised Learning

$\mathcal{X}$ Input set $\mathcal{Y}$ Output set

ρ (unknown) probability on $\mathcal{X} \times \mathcal{Y}$

$\ell : \mathcal{Y} : \mathcal{Y} \mapsto \mathbb{R}$ Loss function

$f : \mathcal{X} \mapsto \mathcal{Y}$ (candidate) input-output predictor

Classification $\mathcal{Y} = \{1, \ldots, K\}$

Regression $\mathcal{Y} \subseteq \mathbb{R}$

Goal: minimise Expected Risk

$$\mathcal{R}(f, \rho) = \mathbb{E}_\rho \, \ell(f(x), y)$$

Supervised Learning

In practice, ρ is **unknown** and it can be only accessed via finite samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$

Learning Algorithm

$$\mathcal{A} : \mathcal{D} \mapsto f$$

Hyper-parameters

Learning rate
Dropout,
iterations …

$$\mathcal{A}(\lambda; \mathcal{D}) = \underset{f \in \mathcal{H}}{\operatorname{argmin}}\ \mathcal{R}(f, \mathcal{D}) + \Omega_\lambda(f)$$

Hypothesis/Inductive Bias

Linear functions
Radial basis functions
Neural Networks …

Empirical Risk $\mathcal{R}(f, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(f\left(x_i\right), y_i\right)$ + Regulariser term e.g. $\Omega_\lambda(f) = \lambda \|f\|^2$

Modern Machine Learning success

POLITECNICO MILANO 1863

AlexNet

$\mathcal{D}$: ImageNet (1.2M images)

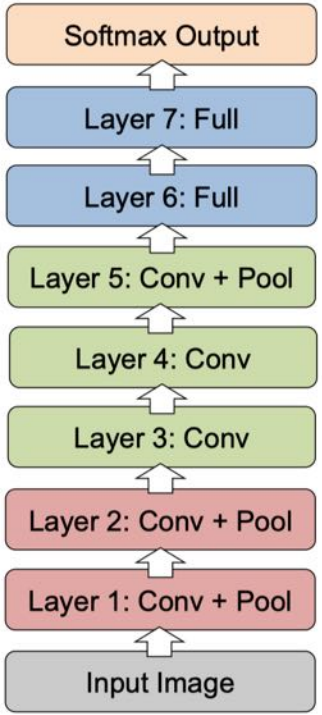
Input set: 224 x 224 images
Output: 1000 classes

$$\underset{\omega}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f\left(x_i;\omega\right), y_i\right) + \lambda\Omega(\omega)$$

$f(x_i;\omega)$ **differentiable**, non-linear function parameterized by ω. 8 layers Neural Network. 60M parameters.

$\ell\left(f\left(x_i;\omega\right), y_i\right)$ cross-entropy loss

$\Omega_\lambda(f)$ L2 regularization, early stopping, dropout



Krizhesky et al. (2012)

Example: image classification

Input set $\mathcal{X} \subseteq \mathbb{R}^{h \times w \times c}$

Output set $\mathcal{Y} = \{1, \ldots, K\}$

class of candidate predictors f_θ

- Learned
- Fixed

$$x \longrightarrow \boxed{\text{Feature Extraction}} \xrightarrow{z} \boxed{f_\theta(z)} \longrightarrow \hat{y}$$

- High dimensional
- raw input

$z \in \mathbb{R}^d$

- Compact representation
- designed to have nice properties

Example: image classification

Input set $\mathcal{X} \subseteq \mathbb{R}^{h \times w \times c}$

Output set $\mathcal{Y} = \{1, \ldots, K\}$

class of candidate predictors f_θ, g_ϕ

- Learned
- Fixed

$$x \longrightarrow \boxed{g_\phi(x)} \xrightarrow{z} \boxed{f_\theta(z)} \longrightarrow \hat{y}$$

$$z \in \mathbb{R}^d$$

Example: image classification

Input set $\mathcal{X} \subseteq \mathbb{R}^{h \times w \times c}$

Output set $\mathcal{Y} = \{1, \dots, K\}$

class of candidate predictors f_θ, g_ϕ

- Learned
- Fixed

Head (actually solving the task)

$$x \longrightarrow \boxed{g_\phi(x)} \xrightarrow{z} \boxed{f_\theta(z)} \longrightarrow \hat{y}$$

Representations learning via Neural Networks
uses back-propagation and gradient descent

Motivations

POLITECNICO
MILANO 1863

1) Theoretical understanding of modern neural architectures

→ Adaptive and Iterative Inference: ResNets as dynamical systems

2) Representations Learning for new data modalities

→ Computer Vision with Asynchronous Event-based data

3) Learning from Limited Labels

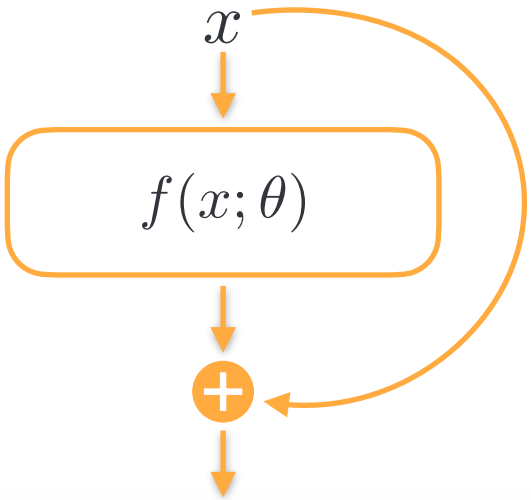
→ Adaptive Representations for One-Shot Video Object Segmentation

Adaptive and Iterative Inference

ResNets as dynamical systems

Residual and Highway Networks

Redesign neural networks to make them **easier to optimize** **even for very large depths** (solving the vanishing gradient problem)



"Shortcut or skip connection"

The gradient can skip layers of computation to assign credit to initial units.

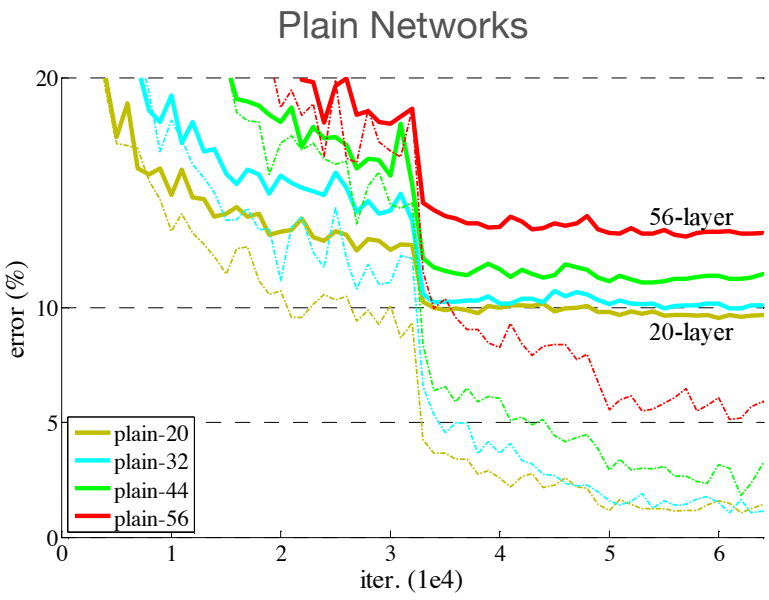
$$y = x + f(x;\theta)$$

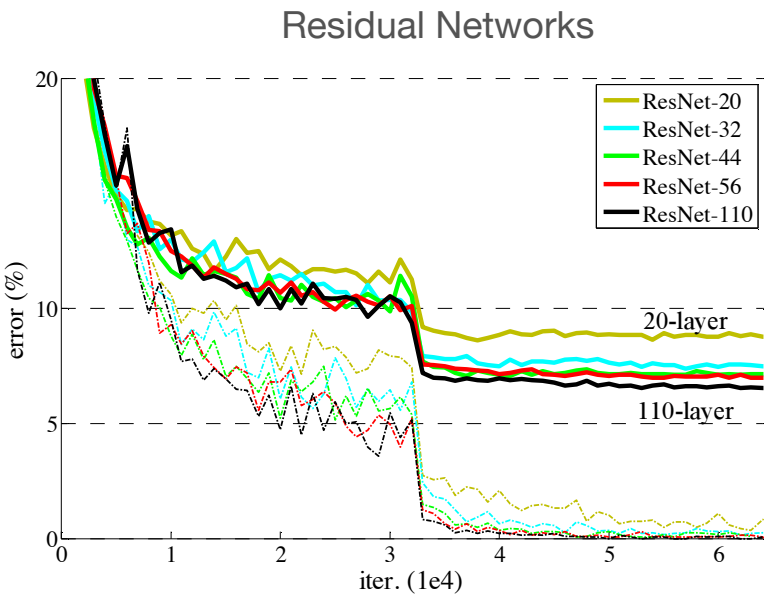
$$y = x \cdot C(x;\theta_C) + f(x;\theta) \cdot T(x;\theta_T)$$

Carry gate Transform gate

He et al. (2015)
Srivastava et al. (2015)

Residual and Highway Networks





He et al. (2015)

Additive Compositional Layers

Most of the successfully trained **very deep architectures** share a **core building block** to compute a vector representation at **layer** $k+1$, parametrised by $\theta(k)$:

$$x(k+1) = x(k) + f(x(k), \theta(k))$$

Previous Representation Additive Non-Linear Transformation

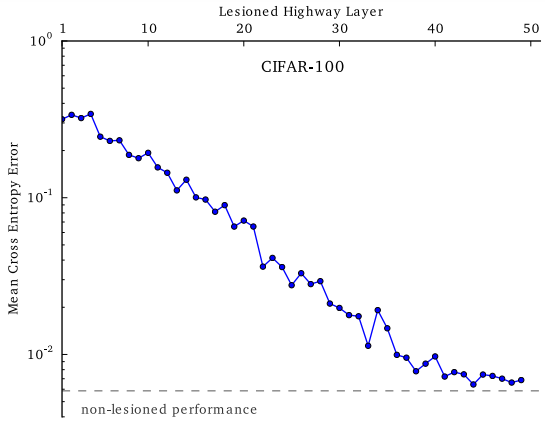
Inductive bias: Iterative inference and features refinement

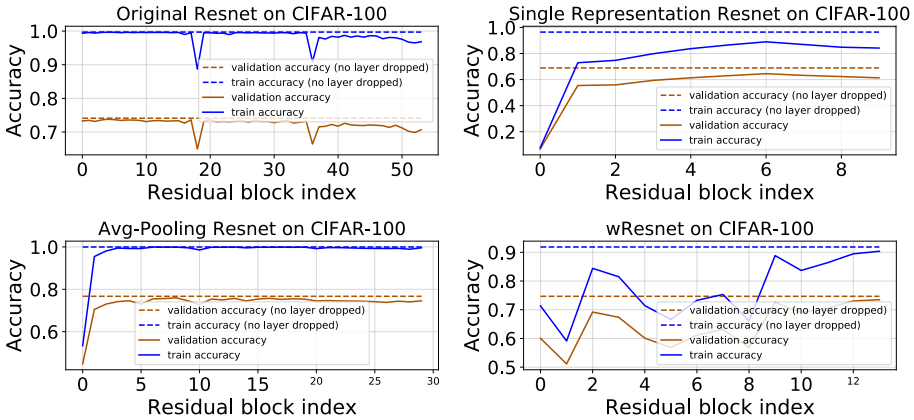
Iterative inference in ResNets and Highway Networks

POLITECNICO MILANO 1863

Lower residual blocks learn **hierarchical representations** (each block discovers a different representation)

Higher residual blocks learn to perform **iterative inference** and **feature refinement.** (keep the semantics of the representation of the previous layer)





Greff et al. (2017)
Jastrzebski et al. (2018)

ResNets as ODEs discretizations

Continuous nonlinear ODE

$$\dot{x} = f(x(t); \Theta)$$

Forward Euler discretization

$$x(k+1) = x(k) + h\, f(x(k); \Theta)$$

ResNets = forward Euler discretization of $\dot{x} = f(x(t); \Theta)$ with step size $h = 1$

Assuming shared weights

$$\theta(k) \equiv \Theta \quad \forall\, 1 \leq k \leq K$$

$$\lim_{h \to 0} \frac{x(k+1) - x(k)}{h} = f(x(k); \Theta)$$

Weinan (2017)

Haber & Ruthotto, (2017)

Deep Networks as ODEs discretizations

Analysis view

Study properties of existing architectures (e.g., ResNet, PolyNet, FractalNet and RevNet) by interpreting them as different **numerical discretizations of ODEs**, (e.g. Backward Euler (implicit) method or Runge-Kutta method)

Synthesis view

Design new architectures from different ODEs and discretizations.

Lu et al. (2018)

Control Theory for Representation Learning

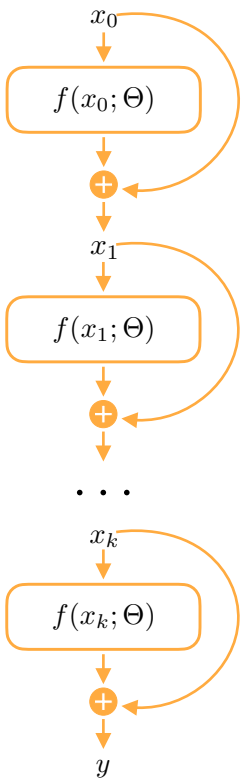
ResNets can be considered discretized dynamical systems that perform iterative inference.

The system (network) should have a stable behaviour such that the forward propagation of the state does not fluctuate.

If we unroll the system to infinity, it should converge to an attractor.

Use Control Theory to study ResNets in terms of stability of their underlined dynamical system.

(Autonomous) Stable ResNet



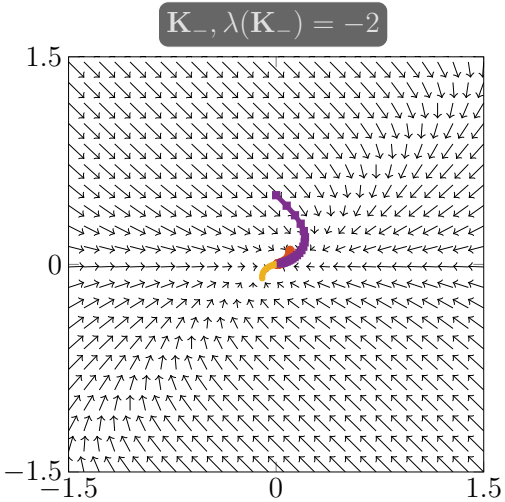
$$x(k+1) = x(k) + f(x(k);\Theta)$$

Assuming shared weights
(time invariant system)

$$\theta(k) \equiv \Theta \quad \forall\, 1 \leq k \leq K$$

ResNet are **autonomous systems.**

Input is connected only to the first layer, acting as **initial condition** x_0.

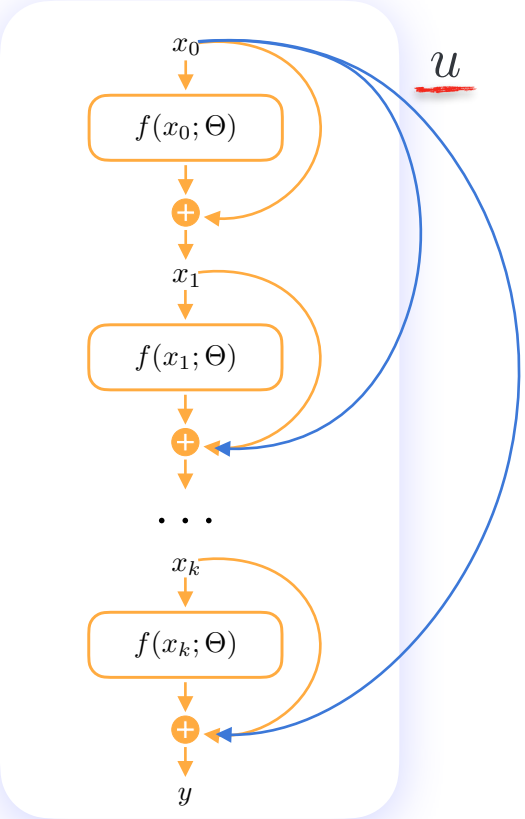


Enforcing stability in **autonomous systems:**

output / information → 0 for all inputs (or trivial solution)

This is useless for Machine Learning applications!

NAIS-Net: Non Autonomous IO-Stable ResNet



Use **input skip connections** to define **Non-Autonomous** Systems.

$$x(k+1) = x(k) + f(x(k), u; \Theta)$$

Assuming shared weights

$$\theta(k) \equiv \Theta \quad \forall\, 1 \leq k \leq K$$

Output trajectories are conditioned on the input and converge to input-dependent attractors.

NAIS-Net block stability

NAIS-Net Fully-Connected block

$$x(k+1) = x(k) + h\sigma(Ax(k) + Bu + b)$$

hidden state matrix

input transfer matrix

Linearization: state-transfer Jacobian for layer k

$$J(x(k), u) = \frac{\partial x(k+1)}{\partial x(k)} = I + h\frac{\partial \sigma(\Delta x(k))}{\partial \Delta x(k)} A$$

Residual Jacobian

Stability Condition (from Lyapunov indirect method)

$$\bar{\rho} := \sup_{(x,u)\in\mathcal{P}} \rho(J(x,u)) < 1$$

spectral radius

where J exists (the nonlinearity σ is not saturated)

Algorithms for Input-Output stability

Algorithm 1 Fully Connected Reprojection

Input: $R \in \mathbb{R}^{\tilde{n} \times n}$, $\tilde{n} \leq n$, $\delta = 1 - 2\epsilon$, $\epsilon \in (0, 0.5)$.

if $\|R^T R\|_F > \delta$ **then**

$\tilde{R} \leftarrow \sqrt{\delta} \frac{R}{\sqrt{\|R^T R\|_F}}$

else

$\tilde{R} \leftarrow R$

end if

Output: $\tilde{R}$

Algorithm 2 CNN Reprojection

Input: $\delta \in \mathbb{R}^{N_C}, C \in \mathbb{R}^{n_X \times n_X \times N_C \times N_C}$, and $0 < \epsilon < \eta < 1$.

for each feature map c **do**

$\tilde{\delta}_c \leftarrow \max\left(\min\left(\delta_c, 1 - \eta\right), -1 + \eta\right)$

$\tilde{C}^c_{i_{\text{centre}}} \leftarrow -1 - \tilde{\delta}_c$

if $\sum_{j \neq i_{\text{centre}}} |C^c_j| > 1 - \epsilon - |\tilde{\delta}_c|$ **then**

$\tilde{C}^c_j \leftarrow \left(1 - \epsilon - |\tilde{\delta}_c|\right) \frac{C^c_j}{\sum_{j \neq i_{\text{centre}}} |C^c_j|}$

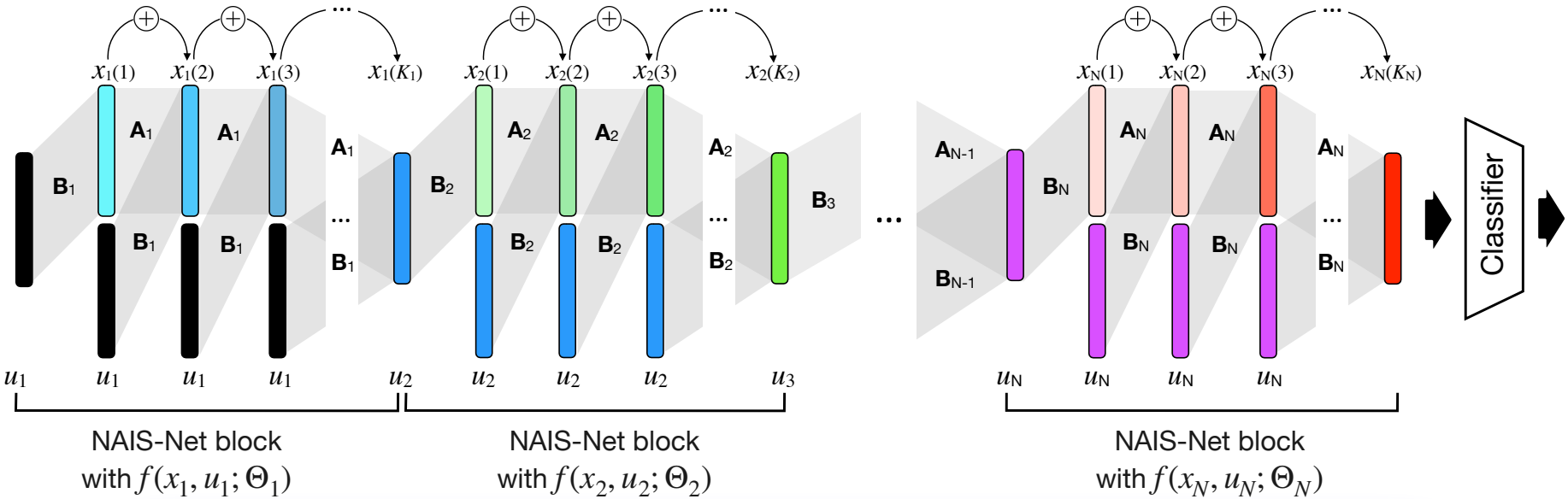
end if

end for

Output: $\tilde{\delta}, \tilde{C}$

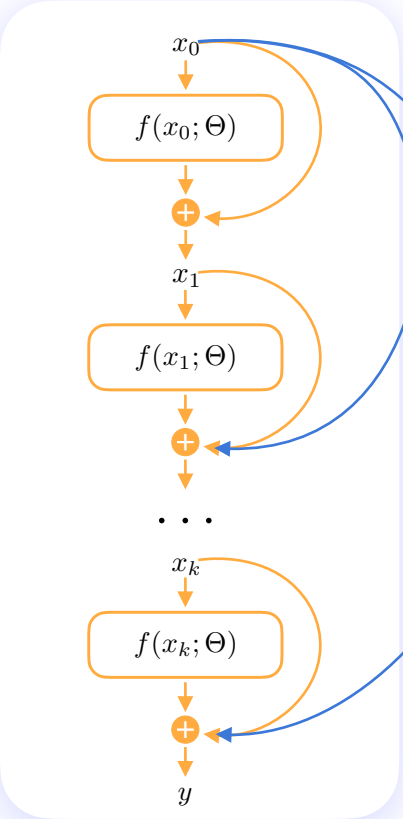
Proposed reprojection algorithms can be used with any **gradient-based optimization** method to **constrain** the parameters in the stability region.

Non-Autonomous Input-Output Stable Architecture



NAIS-Net is a **cascade** of **stable time-invariant** non-autonomous dynamical systems.
The skip connections from the input u_i to all layers in block i make the process non-autonomous.

Pattern-dependent processing depth



Each NAIS-Net block represents an iterative process that models the **trajectories of the input in a different latent space**

$$x(k+1) = x(k) + f(x(k), u; \Theta)$$

Assuming shared weights

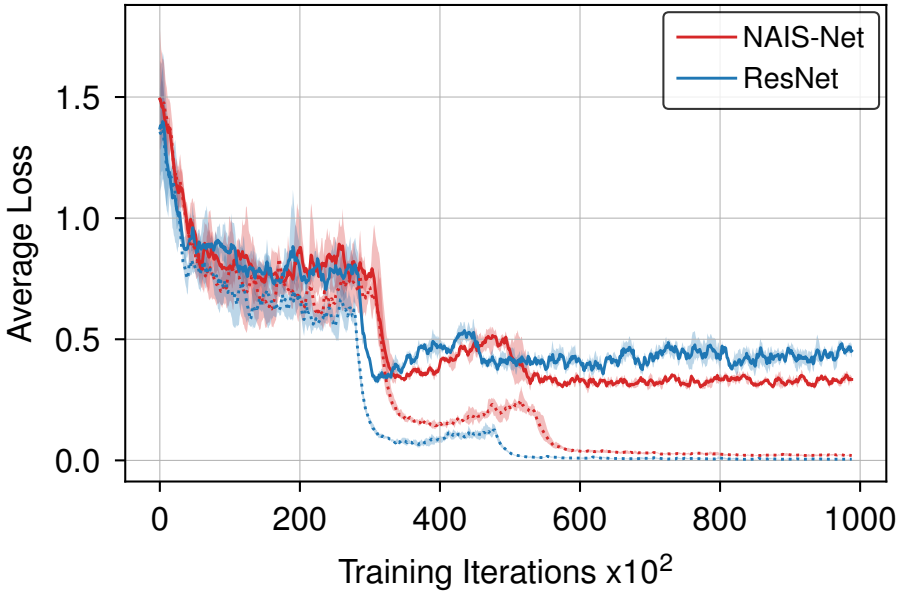
$$\theta(k) \equiv \Theta \;\; \forall\, 1 \leq k \leq K$$

Thanks to stability and shared weights Θ_i NAIS-Net blocks can be unrolled until convergence to input-dependent attractors.

We can then define **stopping criteria** to have a variable **number of processing stages** K_i **conditioned on the input**.

$$\|x(k+1) - x(k)\|_2 < \epsilon$$

Generalization gap on CIFAR-10

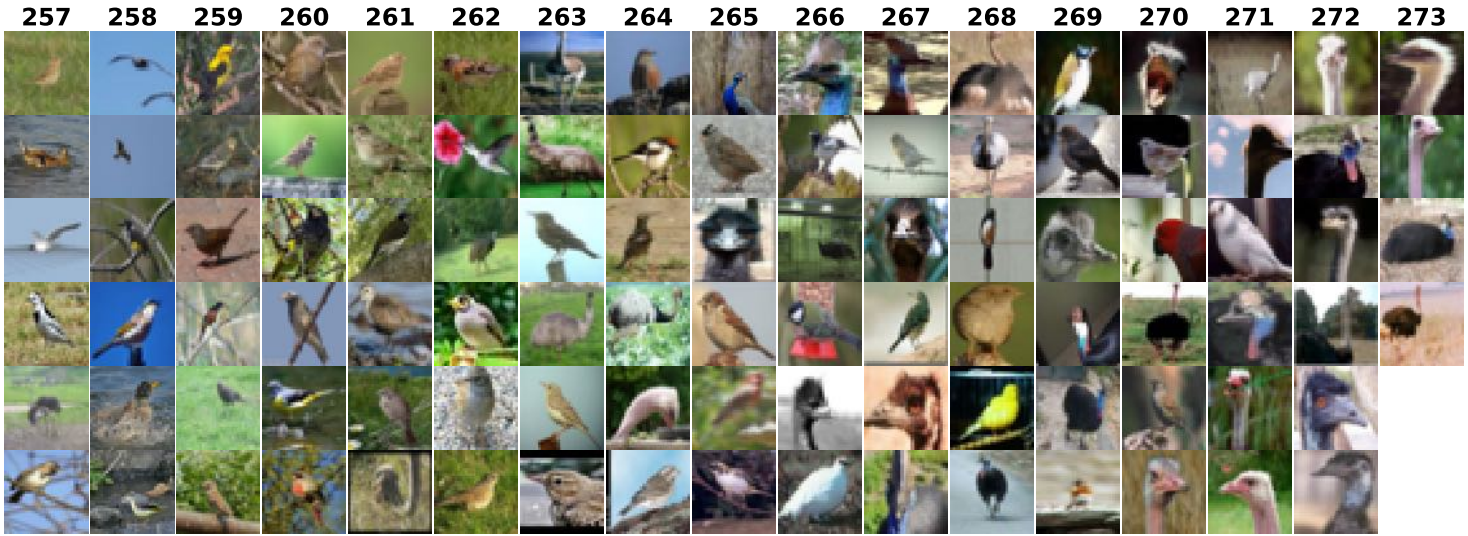


NAIS-Net input-output stability advantages:

- **trajectories are bounded** with respect to noise perturbations increasing robustness and invariance to input perturbations.
- NAIS-Net is less prone to overfitting than a classic ResNet, reducing the generalization gap
- No need of batchnorm, because the forward pass is already well behaved.(we need it only when the input dimensionality changes)

Pattern-dependent processing depth

Final number of layers (depth)



NAIS-Net **adapts processing depth** systematically according to the characteristics of the data.
The depth of the network can be considered as an **additional degree of freedom** of the model.

Pattern-dependent processing depth

Final number of layers (depth)

NAIS-Net **adapts processing depth** systematically according to the characteristics of the data.
The depth of the network can be considered as an **additional degree of freedom** of the model.

Representations Learning for new data modalities

Computer Vision with Asynchronous Event-based data

Event-based Cameras

$$e = (x, y, t, p)$$

Pixel Location

⏱️ Timestamp

Polarity: {-1, +1}

Bio-inspired vision sensors that emulate the functioning of biological retinas.

Smart pixels 🧠

📸 All **independent** from each other

🌟 Only transmit information due to **brightness changes** in the scene

Advantages over conventional cameras:

- ✅ High dynamic range
- ✅ Reduced information redundancy
- ✅ No motion blur
- ✅ Microseconds temporal resolution

Motivation

New data, new challenges

Sparse data → Efficient computation

Asynchronous data → Time integration

$$e = (x, y, t, p)$$

Existing representations do not scale to complex computer vision tasks because:

- ✗ hand-crafted or based on heuristics.
- ✗ and/or disregard spatial correlation.
- ✗ and/or disregard order of arrival

Motivation

POLITECNICO
MILANO 1863

New data, new challenges

Sparse data → Efficient computation

Asynchronous data → Time integration

$$e = (x, y, t, p)$$

Existing representations do not scale to complex computer vision tasks because:

- ✘ hand-crafted or based on heuristics.
- ✘ and/or disregard spatial correlation.
- ✘ and/or disregard order of arrival

Goal: Learn representations from raw events.

Combine the advantages of **event cameras** and those of **frame-based architectures.**

Desiderata

- 🎯 handle sparse data and retain the advantages of asynchronous computation.
- 🎯 preserve spatial information.
- 🎯 end-to-end training with state-of-the-art computer vision systems.

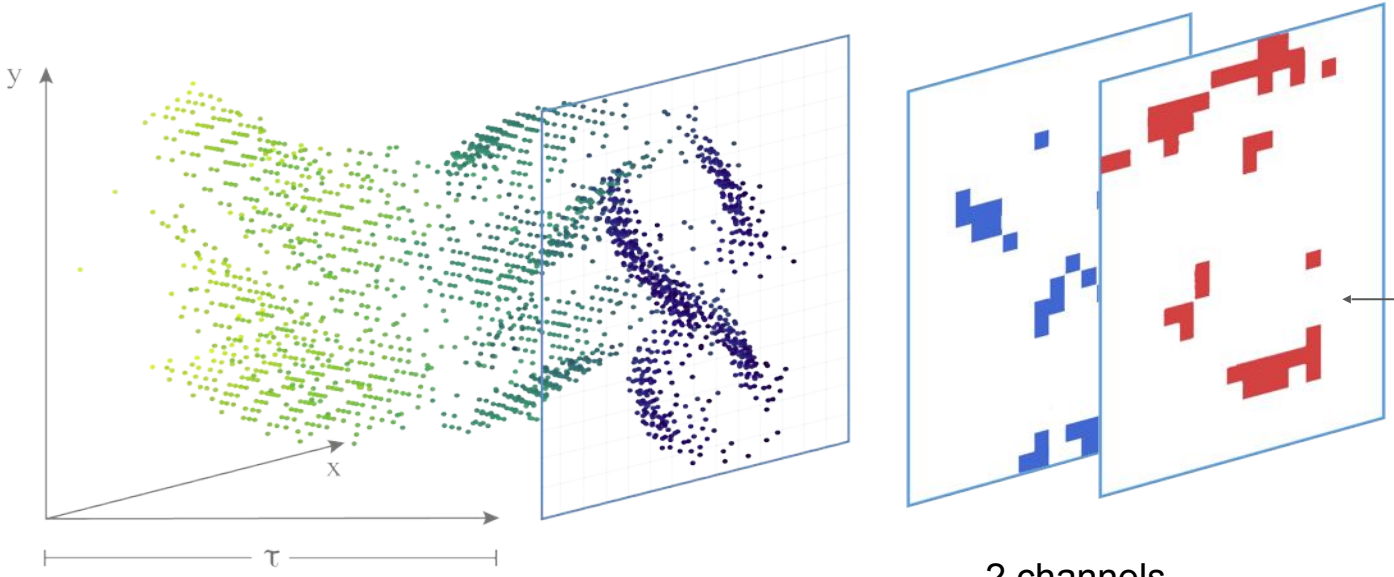
Representations for event-based data

Standard RGB Camera

Event-based Camera

Handcrafted event encoding

Event Representations - Simple Surfaces



2 channels

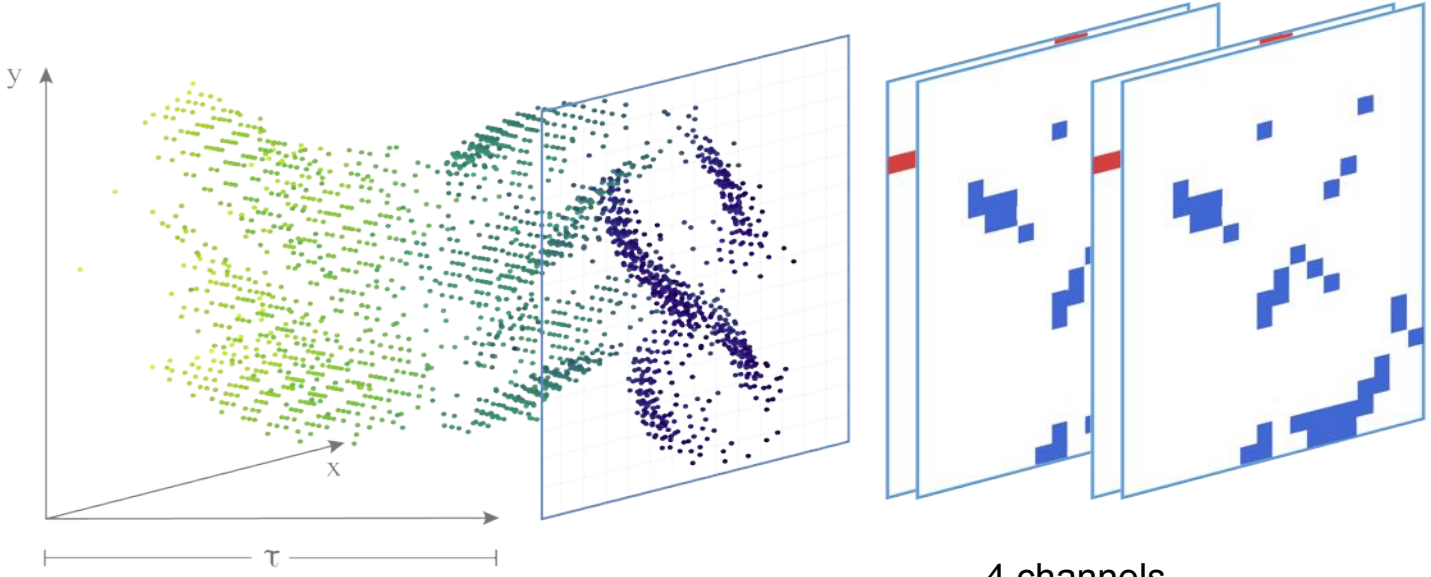
- Negative events

+ Positive events

⏱️ Timestamp

of the last event in each pixel (normalized)

Event Representations - Simple Surfaces



4 channels

- Negative events

+ Positive events

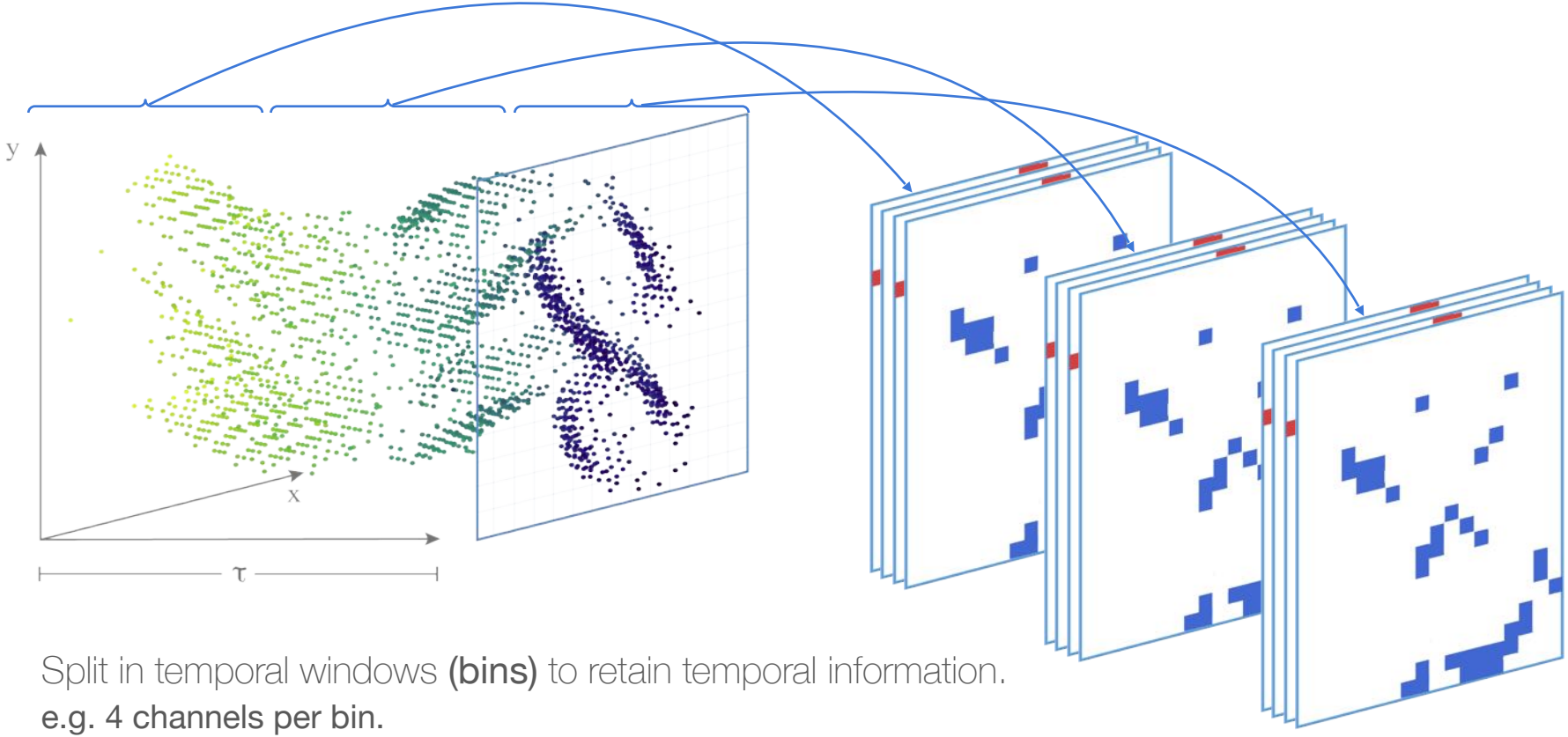
⏱️ Timestamp

of the last event in each pixel

# of events

received in each pixel

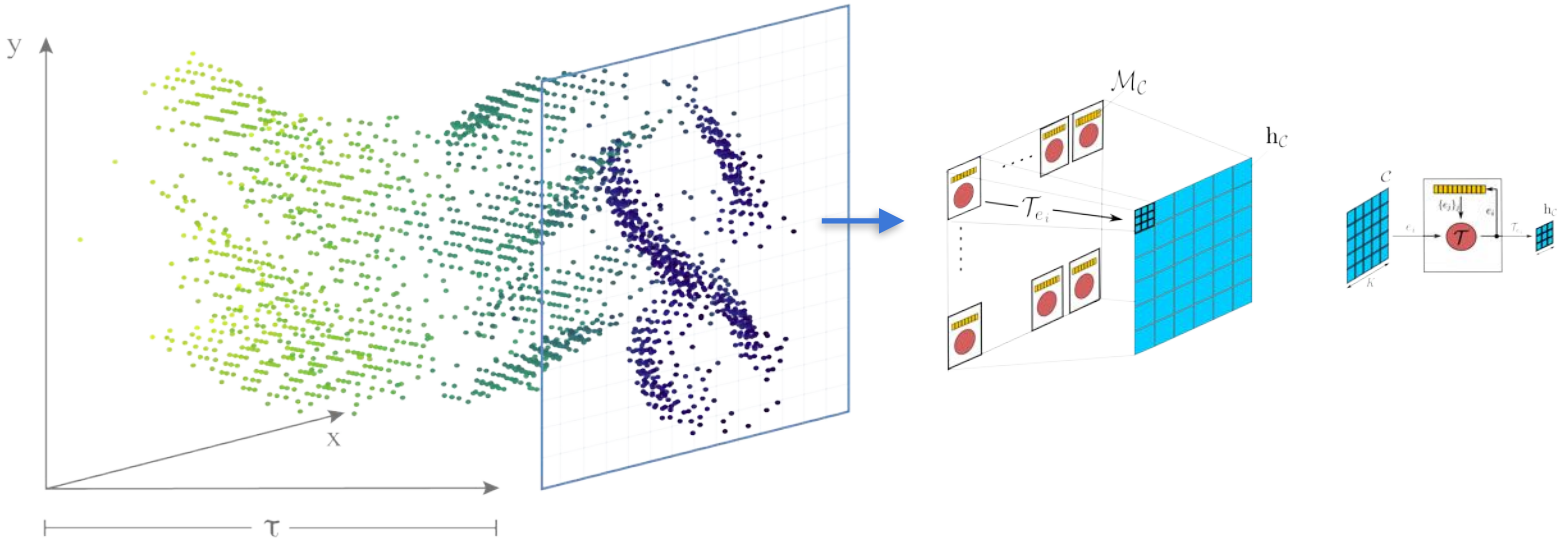
Event Representations - Simple Surfaces



Split in temporal windows **(bins)** to retain temporal information.

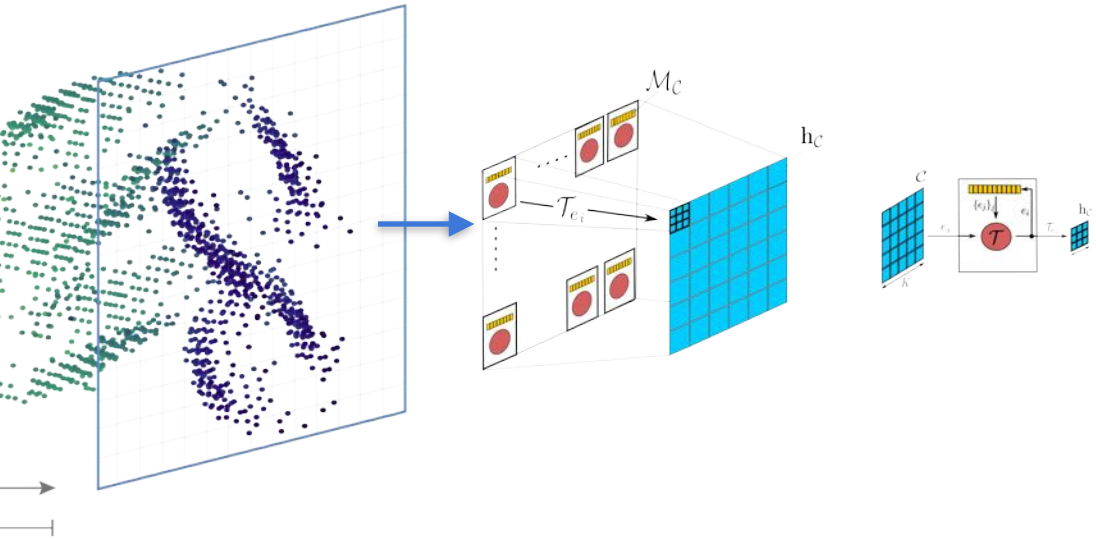
e.g. 4 channels per bin.

Event Representations - HATS



Sironi et al. (2018)
Lagorce et al. (2016)

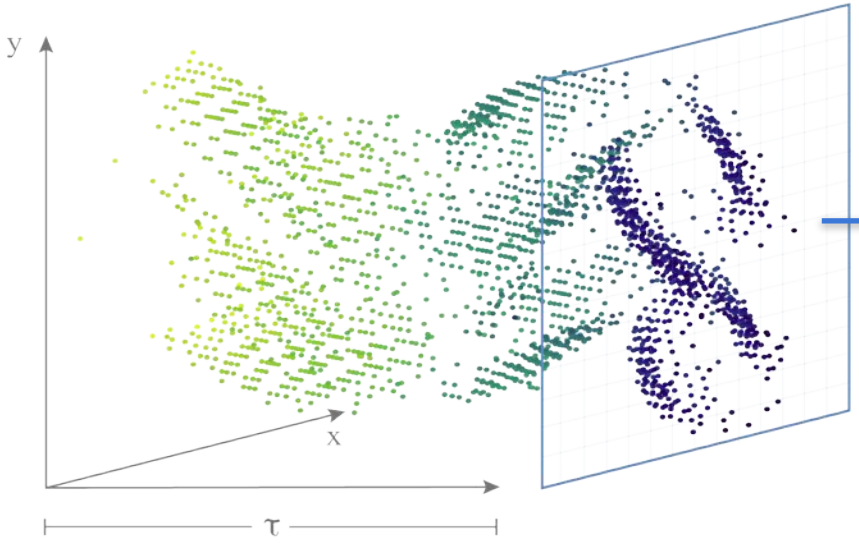
Event Representations - HATS

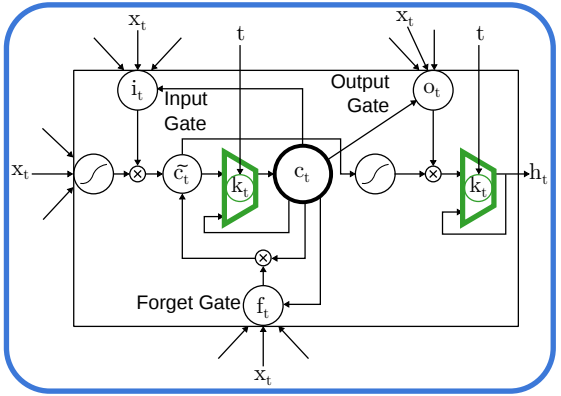


- ✓ Retain temporal information
- ✓ Event features leverage local spatial-temporal patterns
- ✗ Cannot be trained end-to-end and optimized for the task
- ✗ The exponential kernel is hand-crafted
- ✗ The aggregation step is fixed

Sironi et al. (2018)
Lagorce et al. (2016)

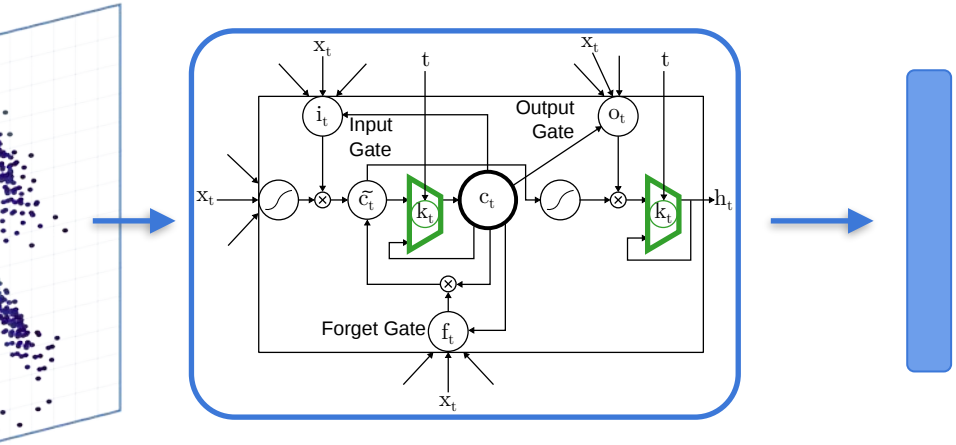
Event Representations - PhasedLSTM





Flattened representation

Event Representations - PhasedLSTM

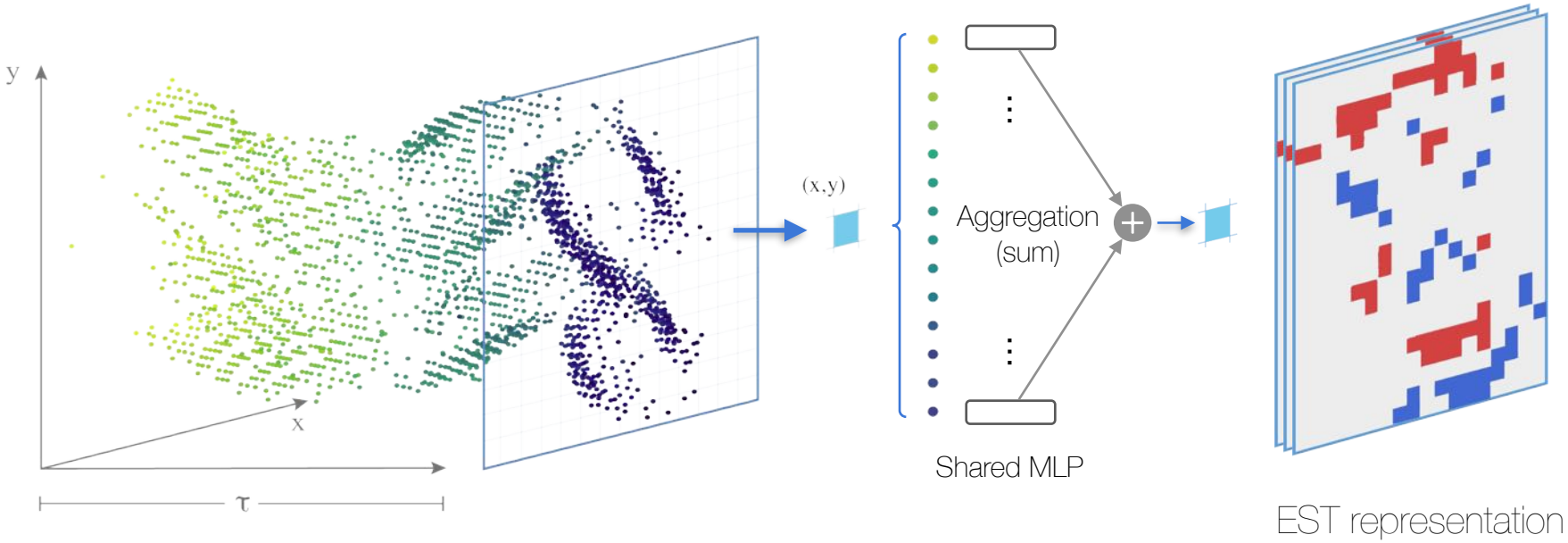


Flattened representation

- ✓ Can be trained end-to-end and optimized for the task
- ✓ The aggregation step is learned
- ✗ Spatial information is lost
- ✗ Does not scale to large frames and complex scenes

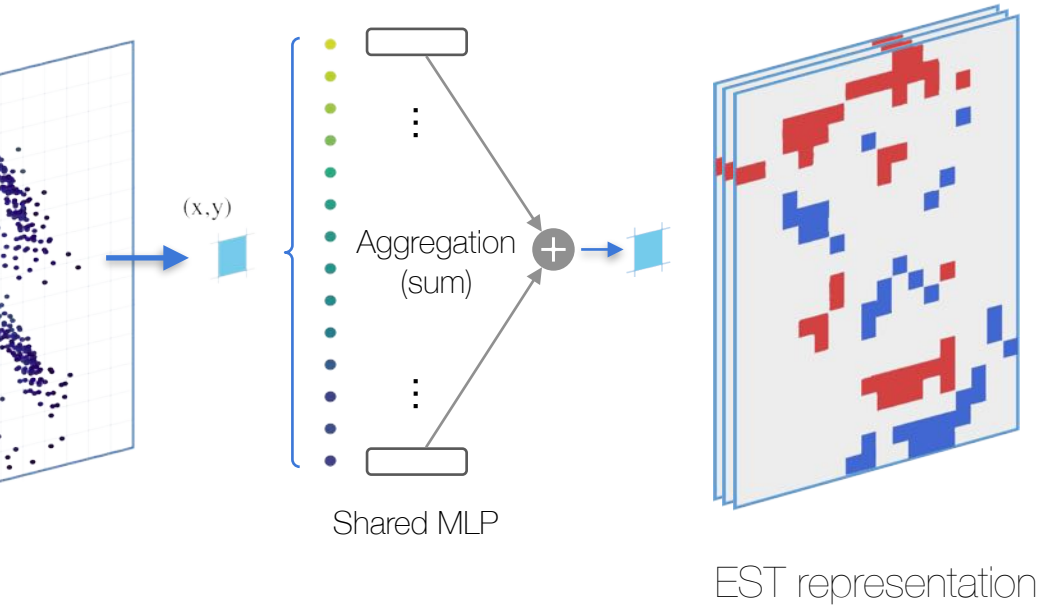
Neil et al. (2016)

Event Representations - EST



Daniel et al. (2019)

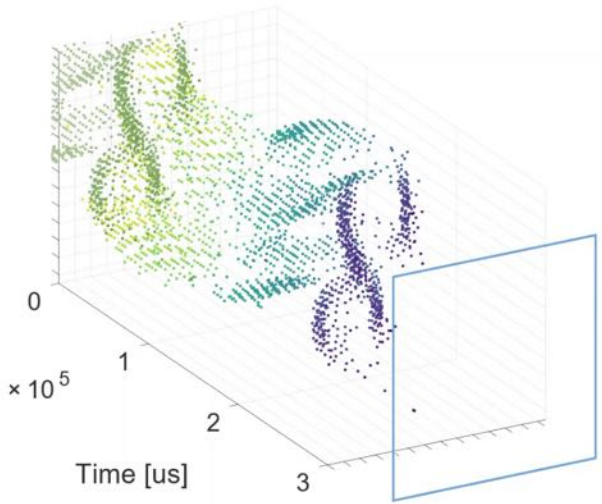
Event Representations - EST



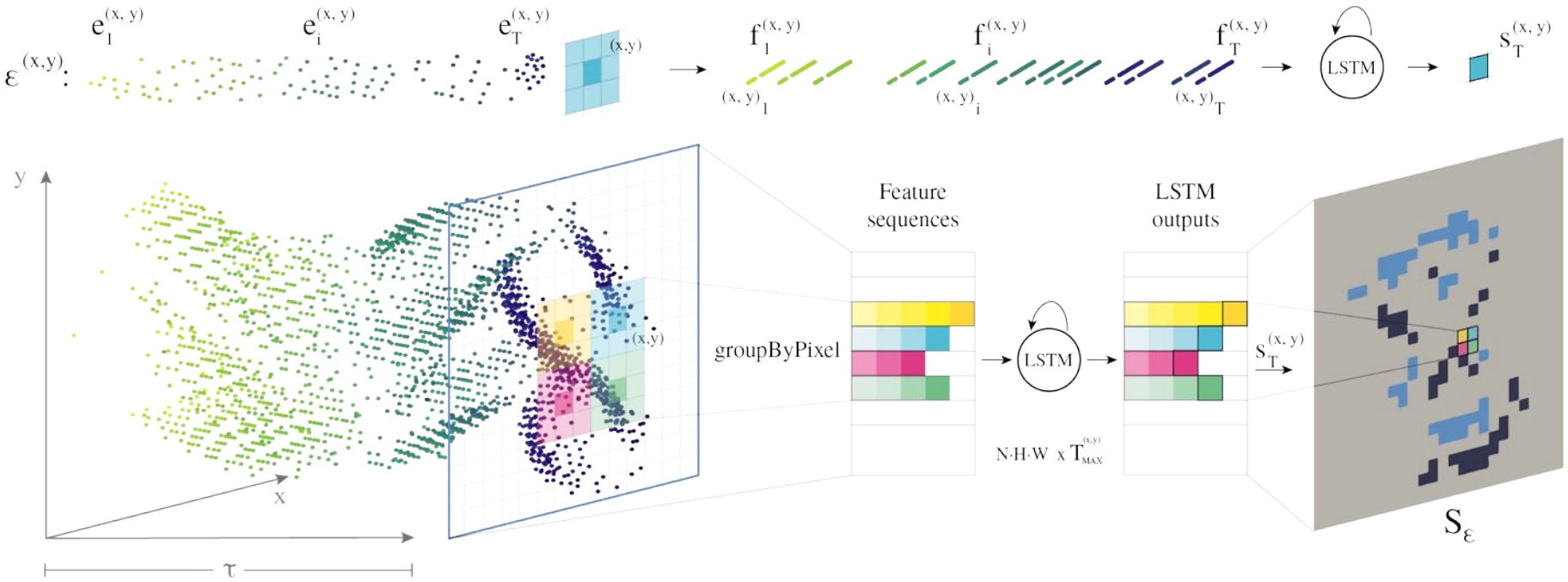
- ✓ Generalize to multiple tasks
- ✓ Can be trained end to end and optimized for the task
- ✗ Original event arrival order is lost
- ✗ Aggregation step is fixed

Daniel et al, 2019

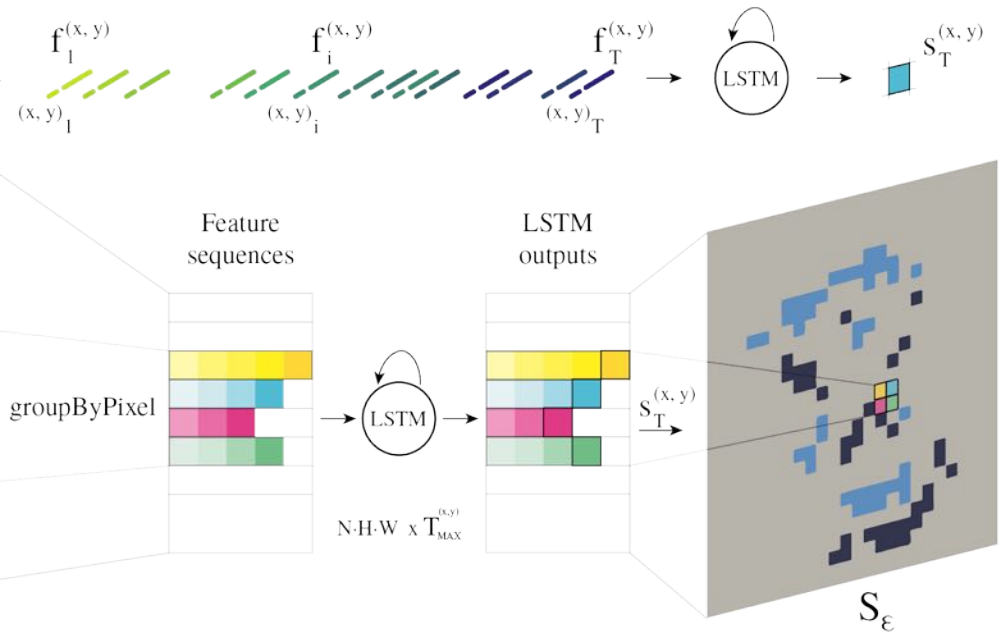
Matrix-LSTM: a Differentiable Recurrent Surface for Events



Matrix-LSTM: a Differentiable Recurrent Surface for Events

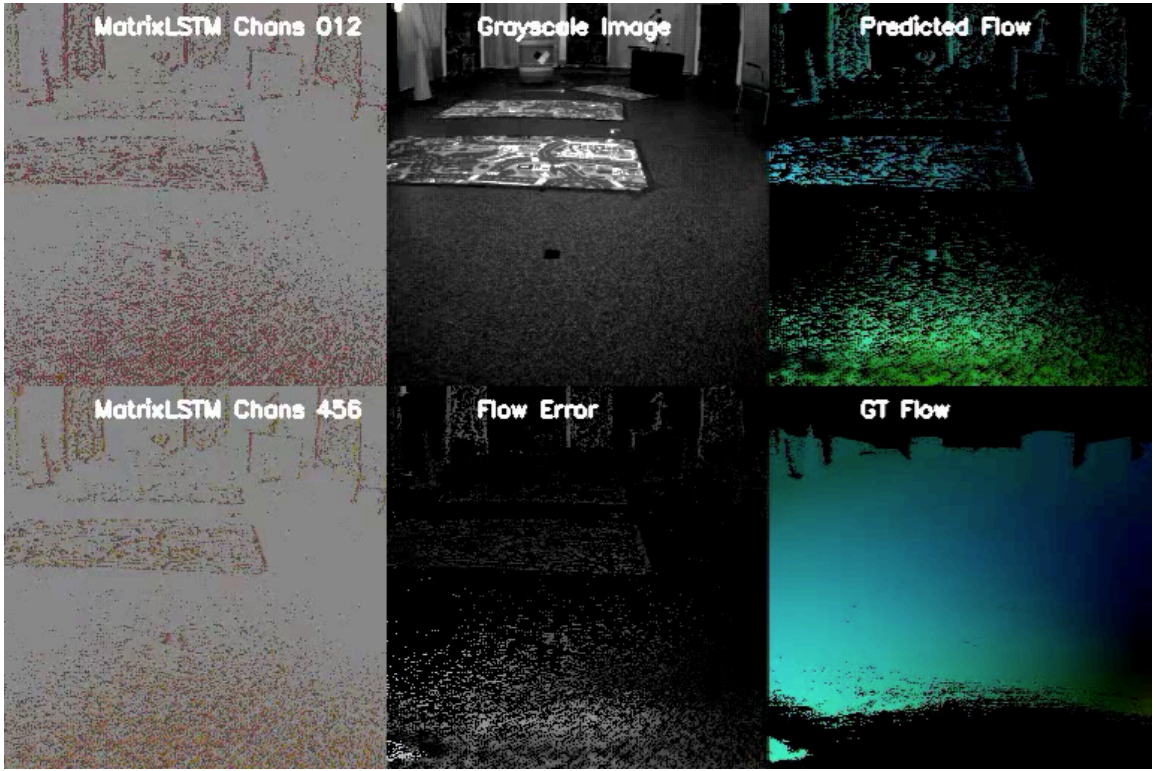


Matrix-LSTM: a Differentiable Recurrent Surface for Events



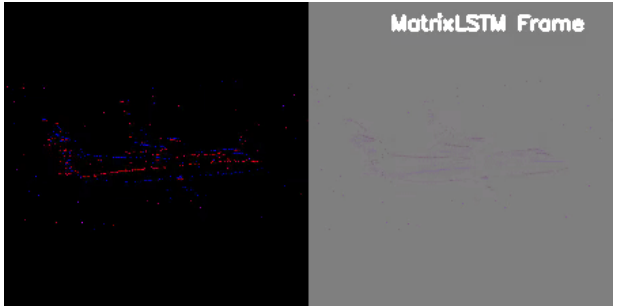
- Events are processed in sequence, reasoning on the previous event dynamics
- The event aggregation mechanism is learned through the **LSTM gates** (how, what and when)
- spatial information is preserved
- It generalizes to multiple tasks
- Can be trained end to end and optimized for the task

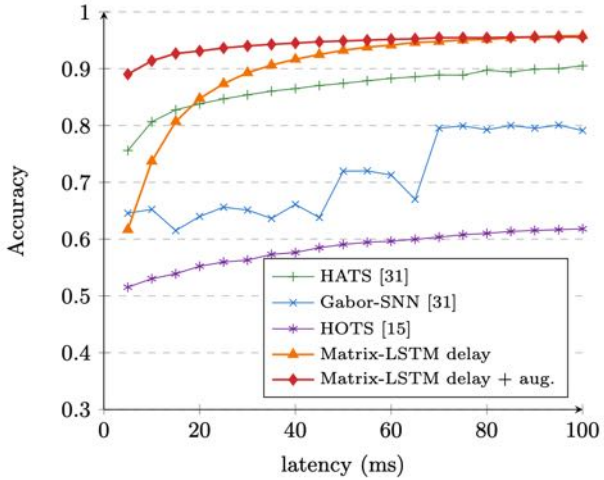
MatrixLSTM: Optical flow prediction



Classification Experiments

Method	Classifier	Channels (bins)	N-Cars	N-Caltech101
H-First [24]	spike-based	-	56.1	0.54
HOTS [15]	histogram similarity	-	62.4	21.0
Gabor-SNN [31]	SVM	-	78.9	19.6
HATS [31]	SVM	-	90.2	64.2
	ResNet34–EST [10]	-	90.9	69.1
	ResNet18–Ev2Vid [26]	-	90.4	70.0
Ev2Vid [26]	ResNet18–Ev2Vid	3	91.0	**86.6**
Matrix-LSTM (Ours)	ResNet18–Ev2Vid	3 (1)	$\mathbf{95.80 \pm 0.53}$	84.12 ± 0.84
	ResNet34–Ev2Vid	3 (1)	$\mathbf{95.65 \pm 0.46}$	85.72 ± 0.37
EST [10]	ResNet34–EST	2 (9)	92.5	81.7
	ResNet34–EST	2 (16)	92.3	83.7
Matrix-LSTM (Ours)	ResNet18–EST	16 (1)	$\mathbf{94.37 \pm 0.40}$	81.24 ± 1.31
	ResNet34–EST	16 (1)	$\mathbf{94.31 \pm 0.43}$	78.98 ± 0.54
	ResNet18–EST	16 (2)	$\mathbf{94.09 \pm 0.29}$	83.42 ± 0.80
	ResNet34–EST	16 (2)	$\mathbf{94.31 \pm 0.44}$	80.45 ± 0.55
	ResNet18–EST	2 (16)	$\mathbf{92.58 \pm 0.68}$	$\mathbf{84.31 \pm 0.59}$
	ResNet34–EST	2 (16)	92.15 ± 0.73	83.50 ± 1.24





Learning from Limited Data

Adaptive Representations for One-Shot Video Object Segmentation

Video Object Segmentation

POLITECNICO
MILANO 1863

New setting, new challenges

- ❗ Segment given objects in the scene. At test time, the first frame annotation is available.
- ❗ No explicit semantic attached to the objects. Foreground/Background.
- ❗ Videos and objects in training set are different from test set!
- ❗ **Generalize to new objects.**

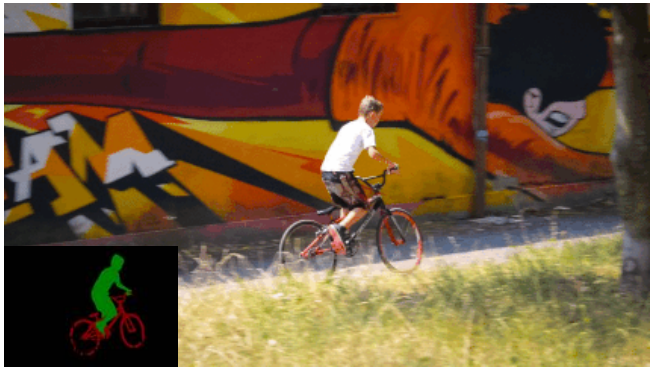
Remember the standard SL setting

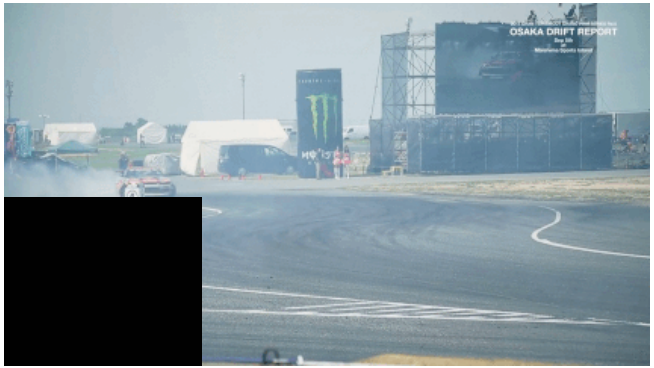
- Fixed number of classes, same at training and test time.
- Generalize to new samples of observed objects.

DAVIS Dataset

Additional Challenges from DAVIS

- ⚠️ Scarce data (only a few videos)
- ⚠️ Ego-motion and occlusions
- ⚠️ Objects change **shape, size** and **perspective** during the video





Supervised Learning Approach

Video Segmentation = Image Segmentation + Time Coherence

Modeling complex motion dynamics (RNNs)
Modeling the concept of objectness (CNNs)

Frame

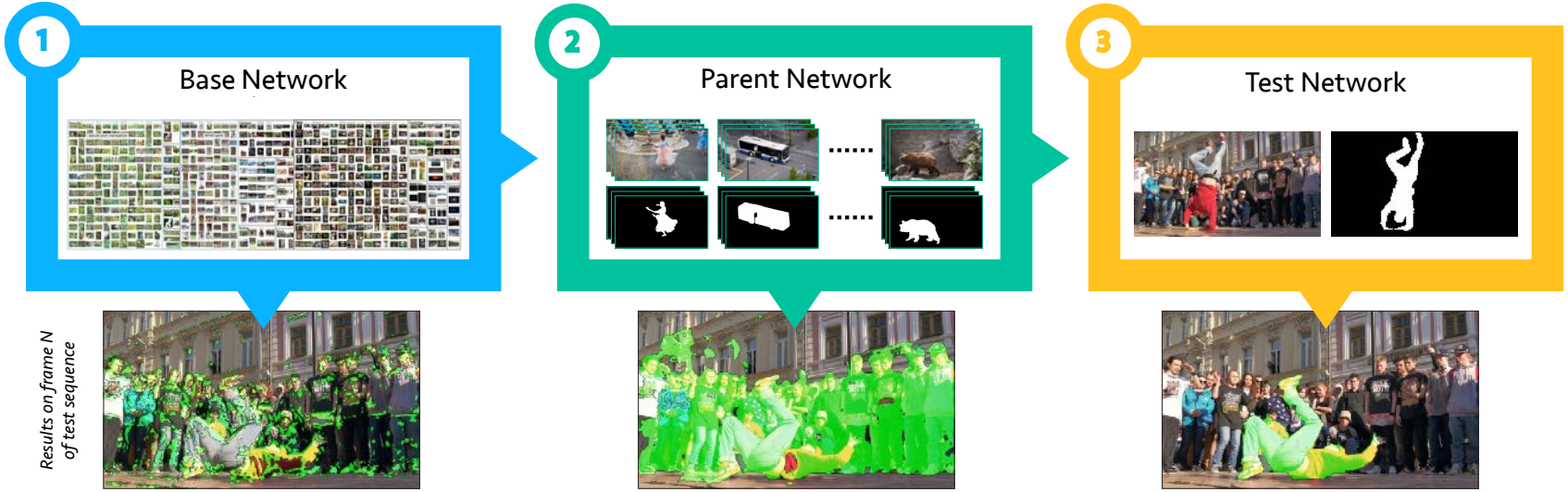
GT

Prediction at epoch 1

Prediction at epoch 10

Complex models with naïve supervised learning approaches quickly overfits

Fine-Tuning approach



Pre-trained on ImageNet

Trained on DAVIS training set

Fine-tuned on 1st frame of test each sequence

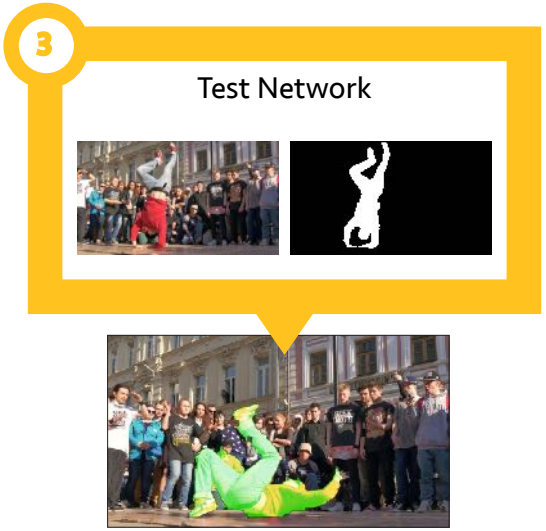
Caelles et al. (2017)

Fine-Tuning approach

POLITECNICO
MILANO 1863

Fine-Tuning each video on single annotation

- ✓ Simple yet effective method
- ✗ **Not efficient:** thousands of optimization steps per video
- ✗ **Hyper-parameters** of the test-time optimization are often excessively handcrafted and fail to generalize between datasets
- ✗ test-time optimization requires **complicated augmentations** to avoid overfitting



Fine-tuned on 1st frame of test each sequence

Caelles et al. (2017)

Meta Learning approach

Meta Learning is an elegant framework that can be used to **extract** and **re-use knowledge** across collections of tasks.

Meta-dataset: collection of episodes (or tasks).

Each video object segmentation can be considered as an independent task.

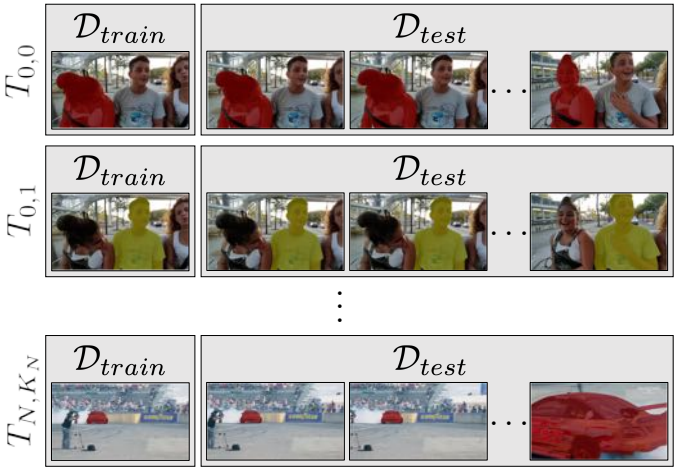
Segmentation task:

- $\mathscr{D}_{train}$: training set of a single example
- $\mathscr{D}_{test}$: test set of next frames

Goal: generalize to other frames of the same video from a single annotation.

Train and test condition match.

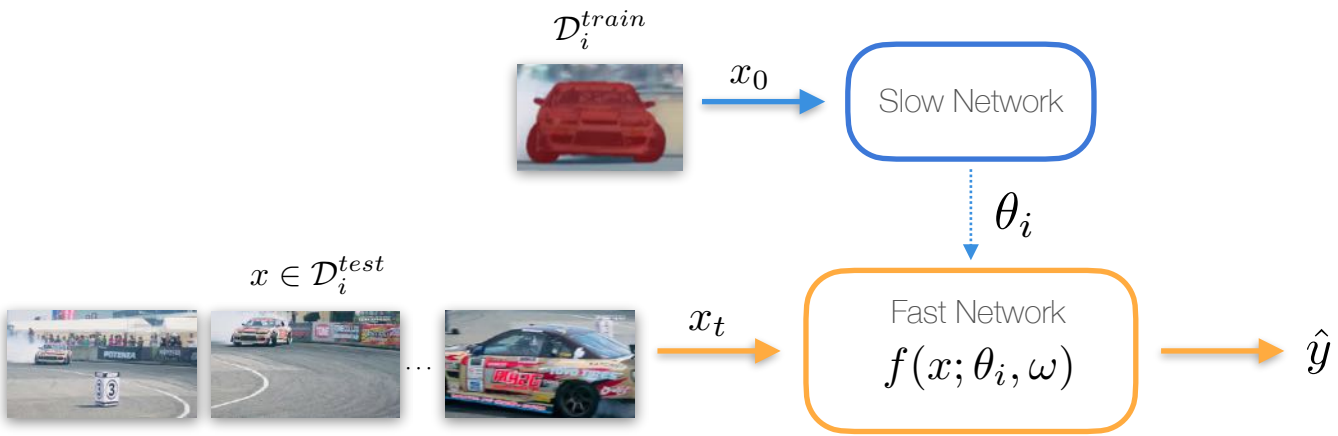
$$\mathcal{T} = \{T_{n,k}\} \text{ with } T_{n,k} = \{\mathcal{D}^{n,k}_{train}, \mathcal{D}^{n,k}_{test}\}$$



Model-based Meta Learning

$$(\omega^*, \phi^*) = \underset{\omega}{\operatorname{argmin}} \sum_{\substack{\tau_i \sim p(\tau) \\ \{\mathcal{D}_i^{train}, \mathcal{D}_i^{val}\} \in \tau_i}} \sum_{(x,y) \in \mathcal{D}_i^{val}} \mathcal{L}^{task}(f(x; \theta_i, \omega), y)$$

$$\text{s.t. } \theta_i = g(\mathcal{D}_i^{train}; \phi)$$

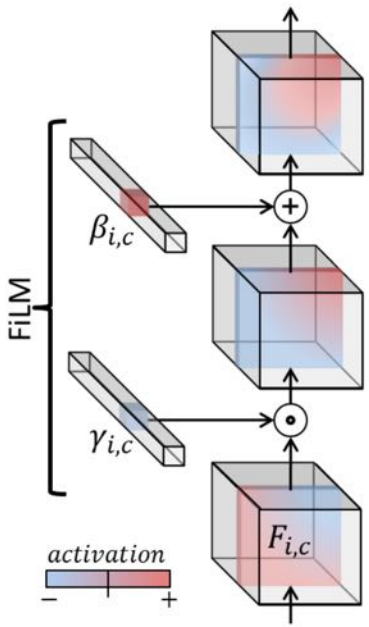


FiLM: Conditioning Layer via Feature Modulation

Generating weights is unfeasible **and prone to overfitting** for large networks.

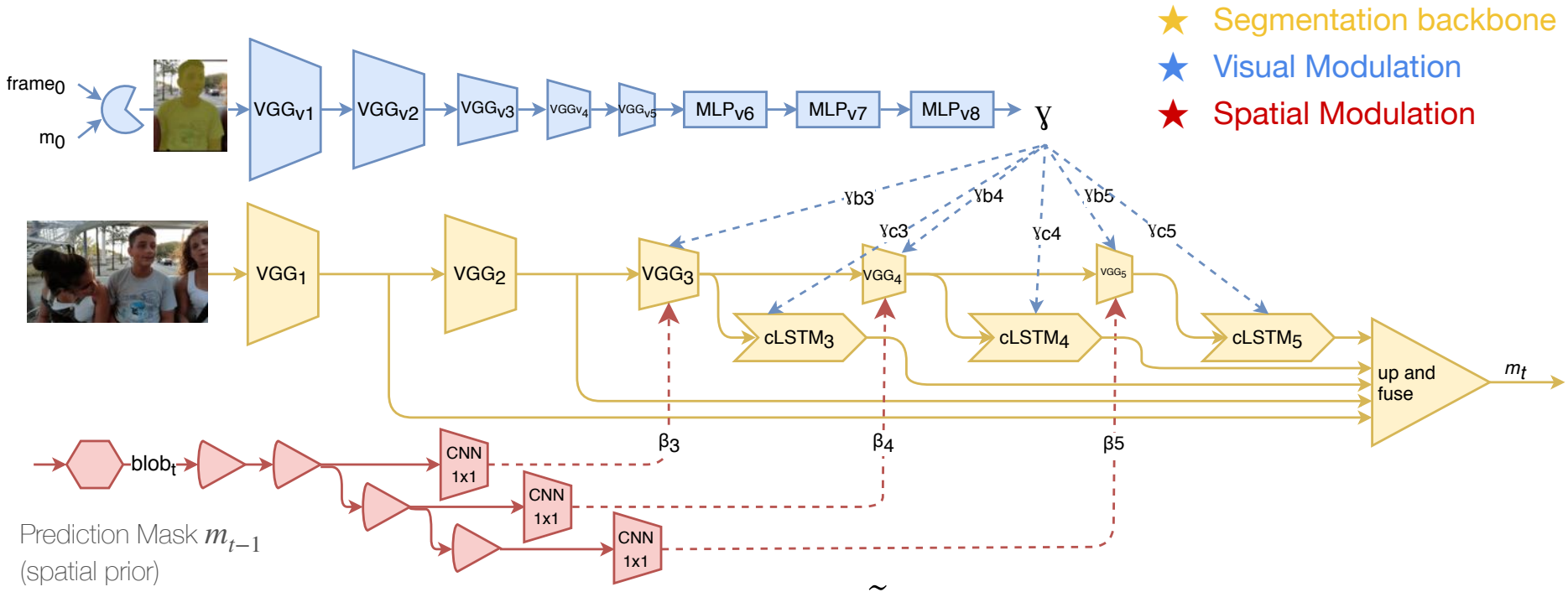
More efficient approach: adapt features representation by conditioning it with task specific affine transformations.

$$\tilde{f}_i = \gamma_i \odot f_i + \beta_i$$



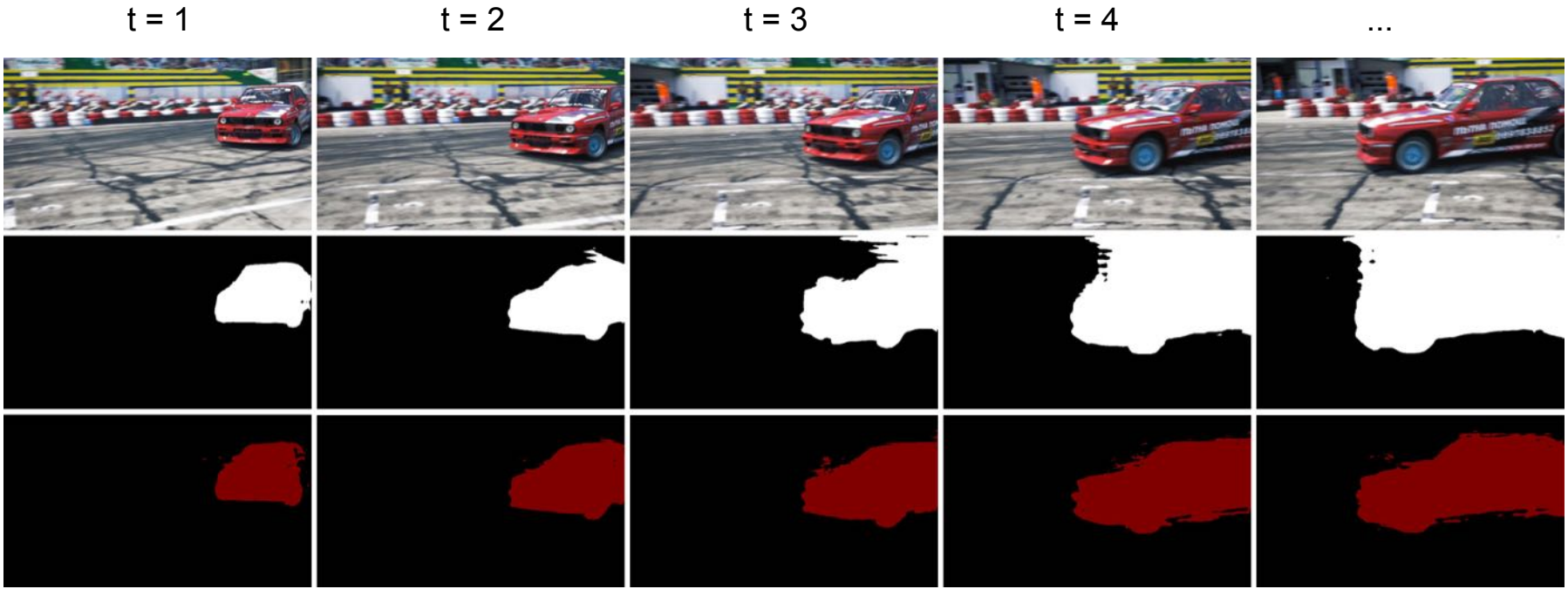
Perez et al. (2017)

Spatio-temporal Features Modulation



$$\tilde{f}_i = \gamma_i \odot f_i + \beta_i(t)$$

Results on single-object segmentation

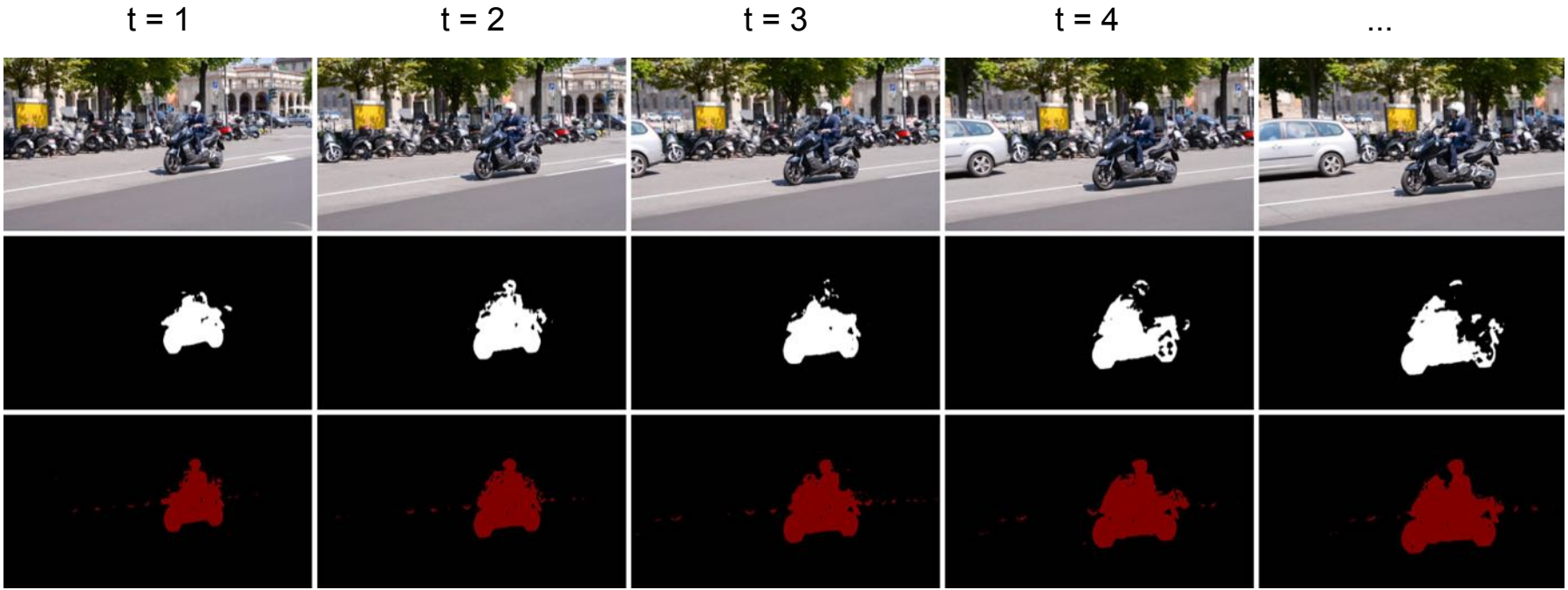


Baseline (OSMN) vs ReConvNet (Ours)

Results on single-object segmentation

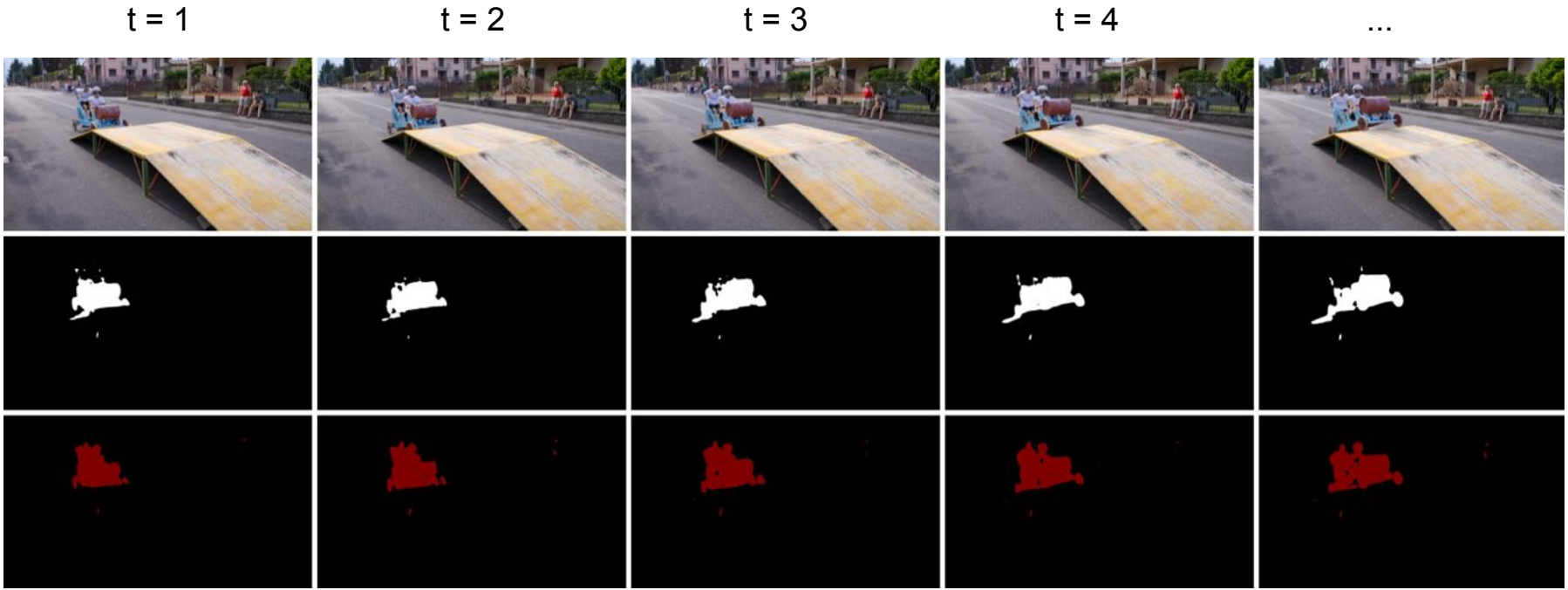
Baseline (OSMN) vs ReConvNet (Ours)

Results on single-object segmentation



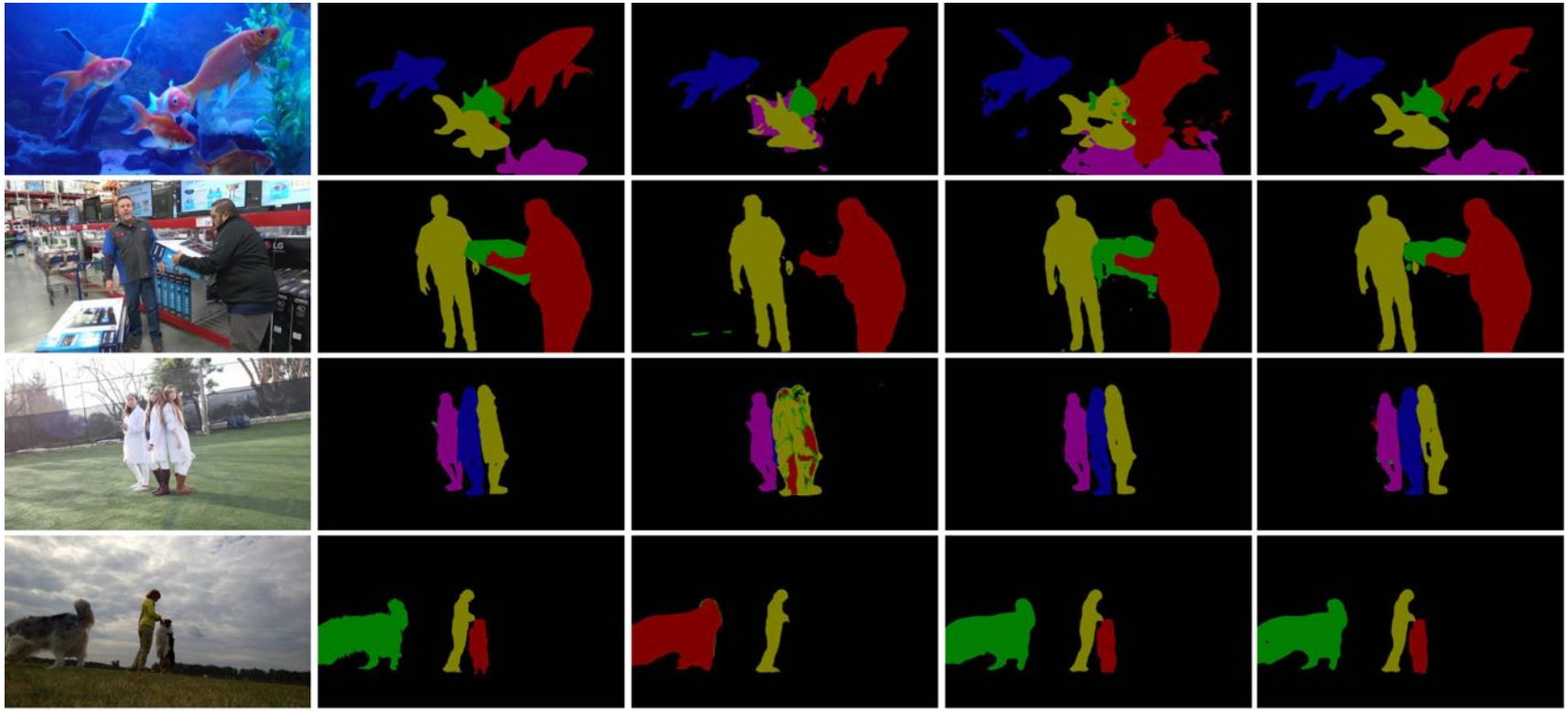
Baseline (OSMN) vs ReConvNet (Ours)

Results on single-object segmentation



Baseline (OSMN) vs ReConvNet (Ours)

Results on multi-objects segmentation



RGB | Ground Truth | OSVOS-S | ReConvNet (ours) | ReConvNet (ours) with FT

Summary and Future Work

POLITECNICO
MILANO 1863

💎 Contributions

💡 **NAIS-Net:** a new neural network with stability guarantees that can be used in safety critical applications. (Ciccone et al. NeurIPS 2018)

🎯 Future work

📌 **NAIS-Net:**

- Adversarial Robustness
- RL applications
- Flow-based generative models

Summary and Future Work

POLITECNICO
MILANO 1863

💎 Contributions

💡 **NAIS-Net:** a new neural network with stability guarantees that can be used in safety critical applications. (Ciccone et al. NeurIPS 2018)

💡 **Matrix-LSTM:** a new general purpose differentiable representation for event-based data that can be used as input for any computer vision task. (Cannici, Ciccone et al. ECCV 2020)

🎯 Future work

📌 **NAIS-Net:**
- Adversarial Robustness
- RL applications
- Flow-based generative models

📌 **Matrix-LSTM:**
- Large scale benchmarks
- Prophesee Automotive Dataset
- New tasks: Object Detection, Depth estimation

Summary and Future Work

POLITECNICO
MILANO 1863

💎 Contributions

💡 **NAIS-Net:** a new neural network with stability guarantees that can be used in safety critical applications. (Ciccone et al. NeurIPS 2018)

💡 **Matrix-LSTM:** a new general purpose differentiable representation for event-based data that can be used as input for any computer vision task. (Cannici, Ciccone et al. ECCV 2020)

💡 **ReConvNet:** an efficient method for Video Object Segmentation that can adapt its representation at test time to new objects with a single annotated frame. (Lattari*, Ciccone* et al CVPRW 2018)

🎯 Future work

📌 **NAIS-Net:**
- Adversarial Robustness
- RL applications
- Flow-based generative models

📌 **Matrix-LSTM:**
- Large scale benchmarks
- Prophesee Automotive Dataset
- New tasks: Object Detection, Depth estimation

📌 **ReConvNet:**
- Improve Object localization and tracking
- Cross-domain One Shot Learning
- Gradient-based adaptation

Collaborators 🙌 🙏

Marco
Gallieri

Jonathan
Masci

Christian
Osendorfer

Faustino
Gomez

Marco
Cannici

Andrea
Romanoni

Matteo
Matteucci

Francesco
Visin

Francesco
Lattari

Publications

POLITECNICO
MILANO 1863

★ **Marco Ciccone**, Marco Gallieri, Jonathan Masci, Christian Osendorfer, Faustino Gomez (2018).
"**NAIS-Net: Stable Deep Networks from Non Autonomous Differential Equations.**"
In: *Advances in Neural Information Processing Systems* ***(NeurIPS)***

★ Marco Cannici, **Marco Ciccone**, Andrea Romanoni, Matteo Matteucci (2020).
"**A Differentiable Recurrent Surface for Asynchronous Event-Based Data.**"
In: *Proceedings of the European Conference on Computer Vision* ***(ECCV)***

★ Lattari*, **Ciccone***, Jonathan Masci, Matteo Matteucci, Francesco Visin (2018).
"ReConvNet: Video Object Segmentation with Spatio-Temporal Features Modulation."
In: *DAVIS Challenge on Video Object Segmentation - CVPR Workshops*

- Cacciamani Federico, Andrea Celli, **Marco Ciccone**, and Nicola Gatti (2021).
"Multi-Agent Coordination in Adversarial Environments through Signal Mediated Strategies."
In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems* ***(AAMAS)***

- Cannici Marco, **Marco Ciccone**, Andrea Romanoni, and Matteo Matteucci (2019a).
"Attention Mechanisms for Object Recognition with Event-Based Cameras."
In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision* ***(WACV)***

- Cannici Marco, **Marco Ciccone**, Andrea Romanoni, and Matteo Matteucci (2019b).
"Asynchronous Convolutional Networks for Object Detection in Neuromorphic Cameras."
In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* - **(Best Paper Award).**

- Romanoni Andrea, **Marco Ciccone**, Francesco Visin, and Matteo Matteucci (2017).
"Multi-View Stereo with Single-View Semantic Mesh Refinement."
In: *Proceedings of the IEEE International Conference on Computer Vision Workshops, Reconstruction Meets Semantic, (ICCVW).*

POLITECNICO
MILANO 1863

🙋‍♂️ Thank you!

Questions?