Feature Space Optimization for Semantic Video Segmentation

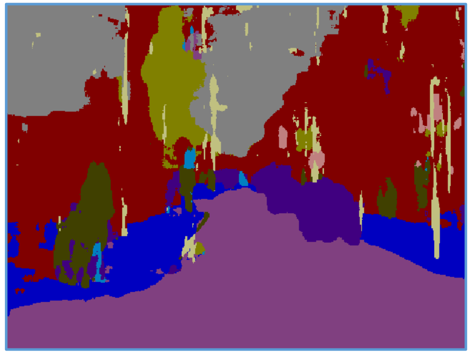
Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun

Objective

Semantic image segmentation

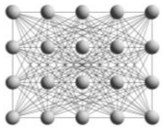
Independent prediction

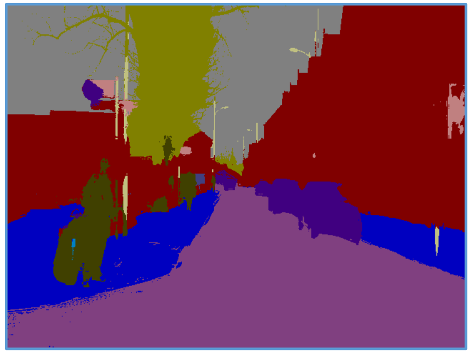
e.g., ConvNet



Structured prediction

e.g., DenseCRF



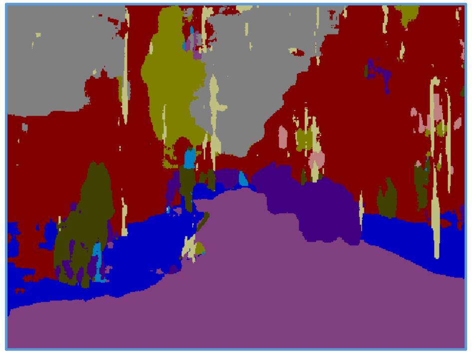


Objective

Semantic image segmentation

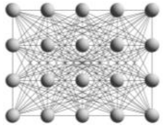
Independent prediction

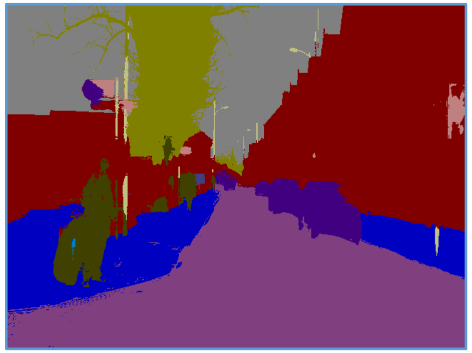
e.g., ConvNet



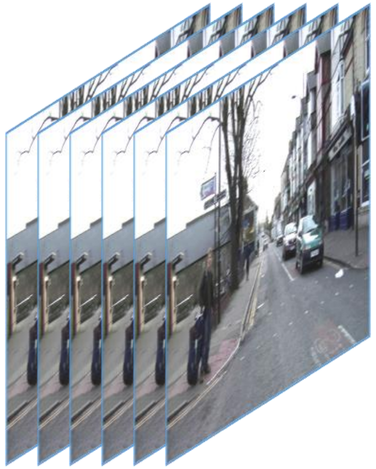
Structured prediction

e.g., DenseCRF





Semantic *video* segmentation:

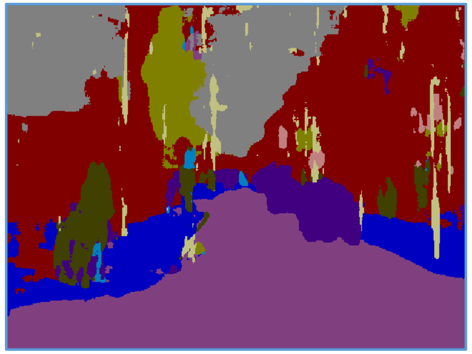


Objective

Semantic image segmentation

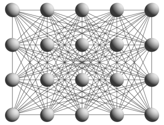
Independent prediction

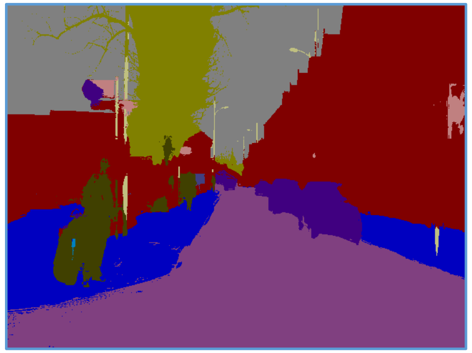
e.g., ConvNet



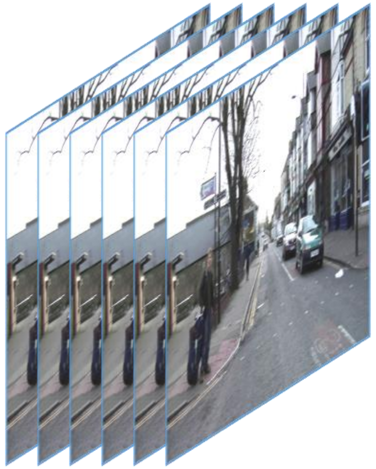
Structured prediction

e.g., DenseCRF

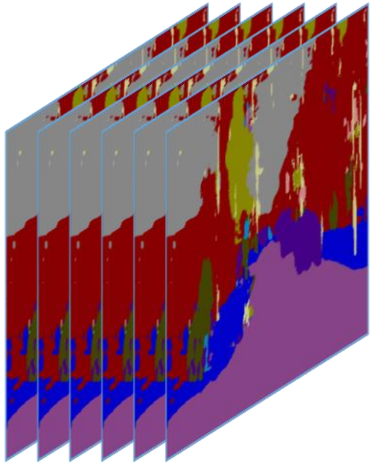




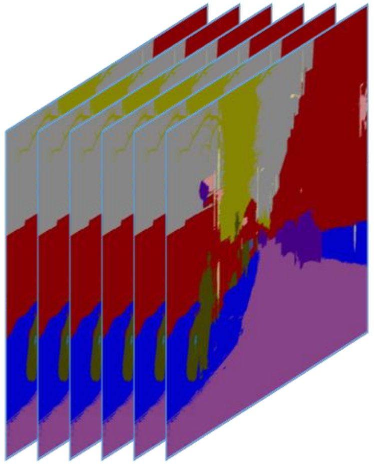
Semantic *video* segmentation: **Naive approach**



Per-frame independent prediction



Per-frame structured prediction

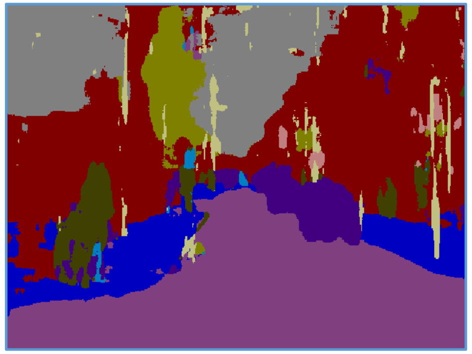


Objective

Semantic image segmentation

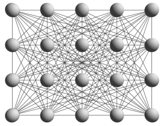
Independent prediction

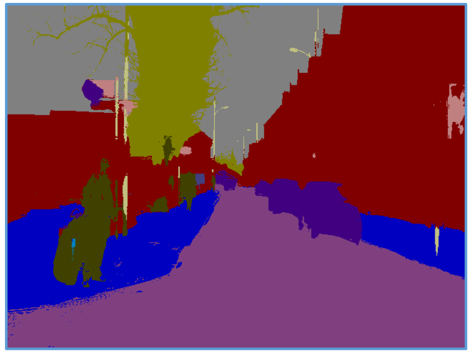
e.g., ConvNet



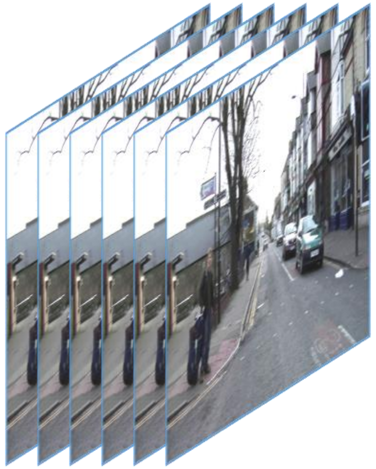
Structured prediction

e.g., DenseCRF

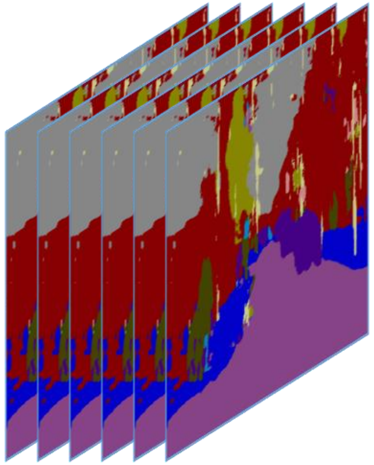




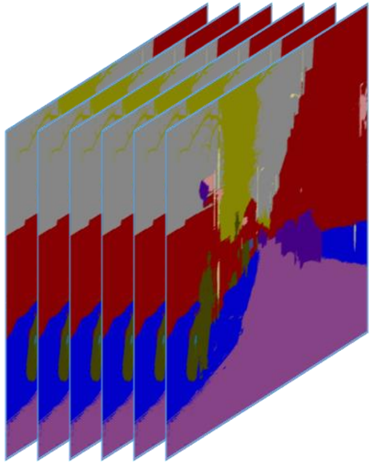
Semantic *video* segmentation: **Our approach**



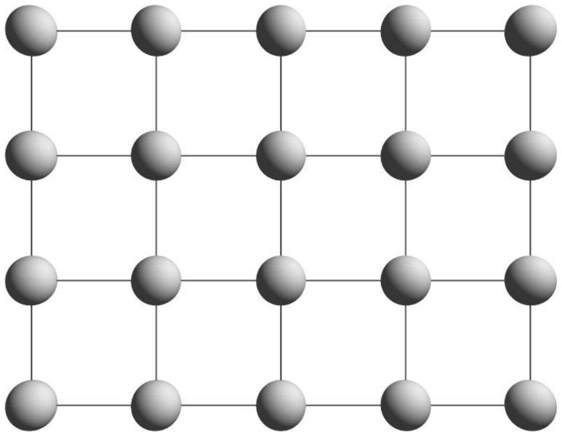
Per-frame independent prediction



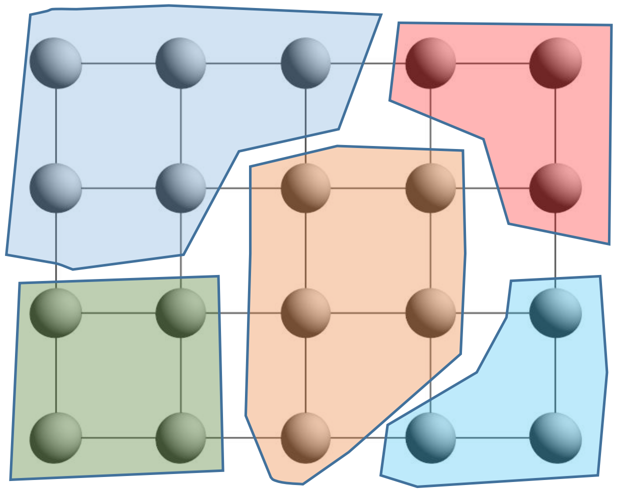
Structured prediction for the entire video



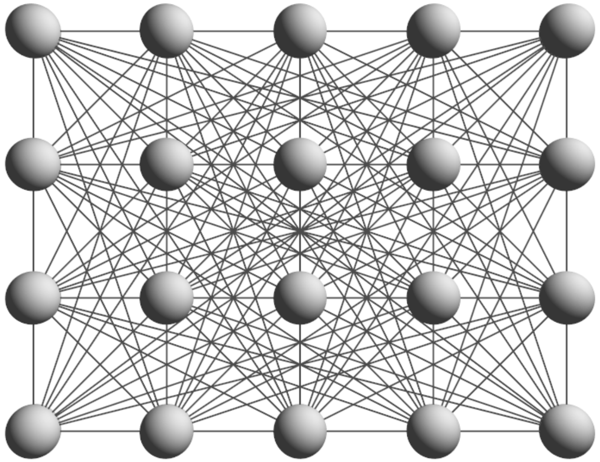
Structured Prediction



Grid CRF

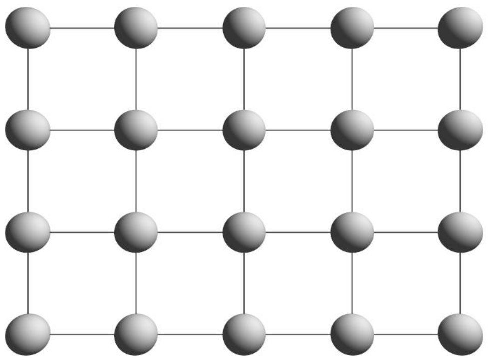


Higher-order CRF

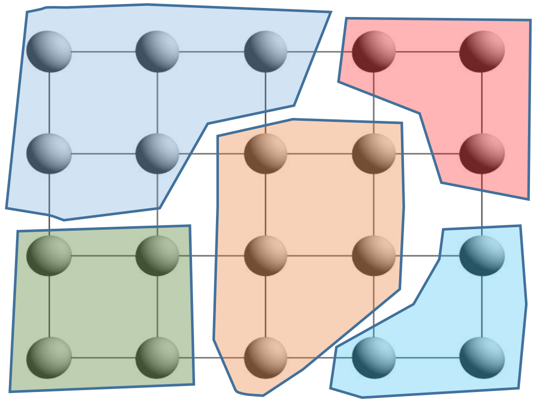


Dense CRF

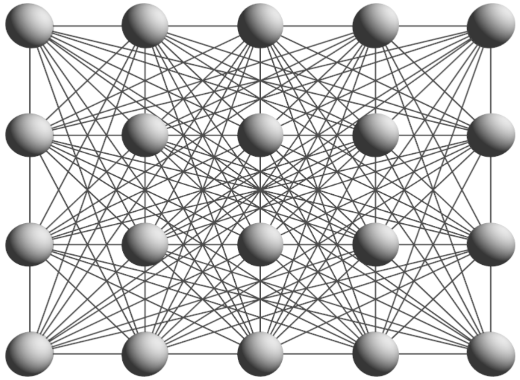
Structured Prediction



Grid CRF

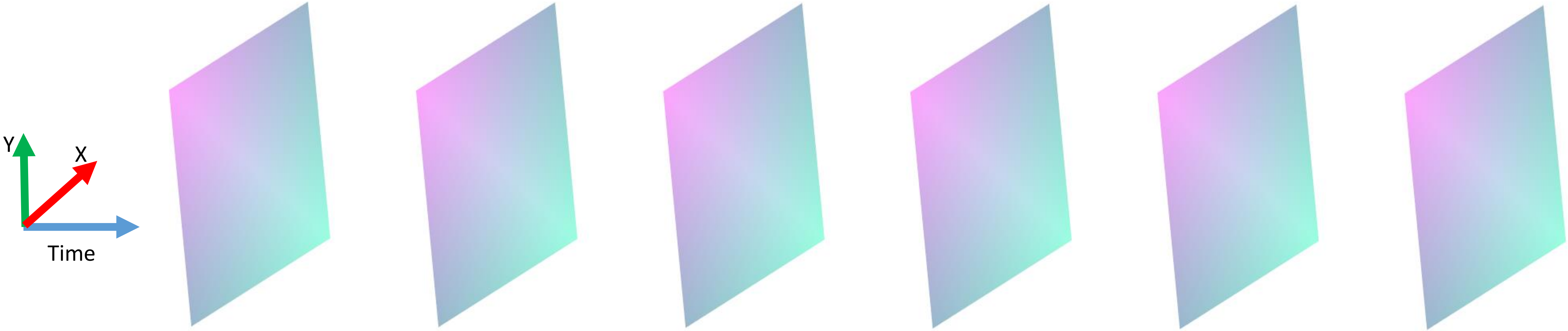


Higher-order CRF

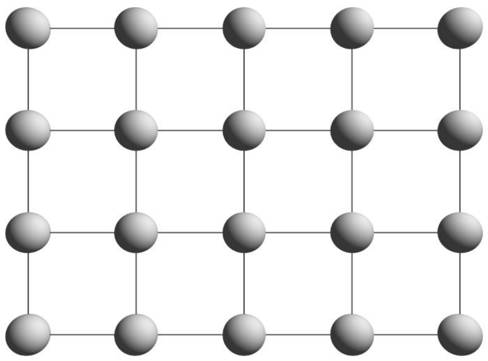


Dense CRF

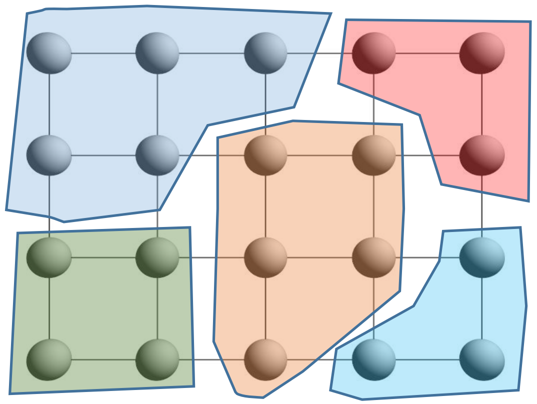
What about video?



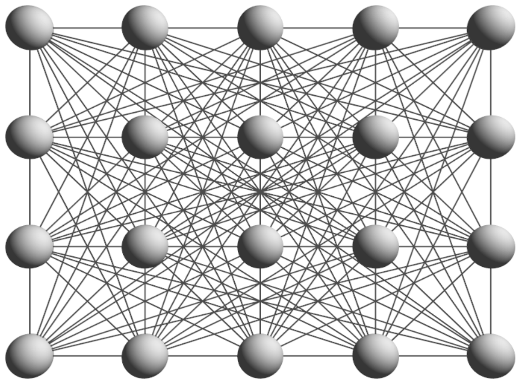
Structured Prediction



Grid CRF

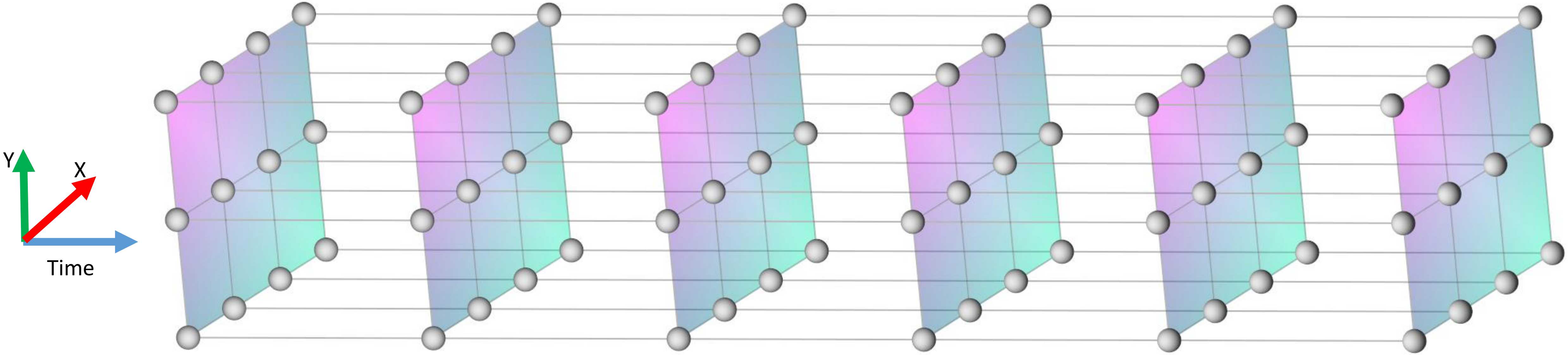


Higher-order CRF

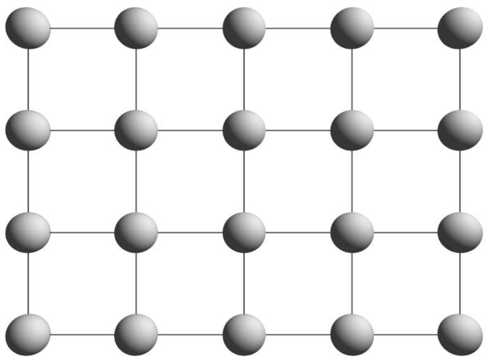


Dense CRF

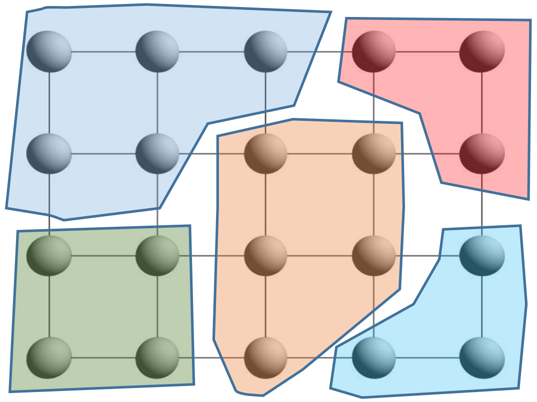
3D grid CRF



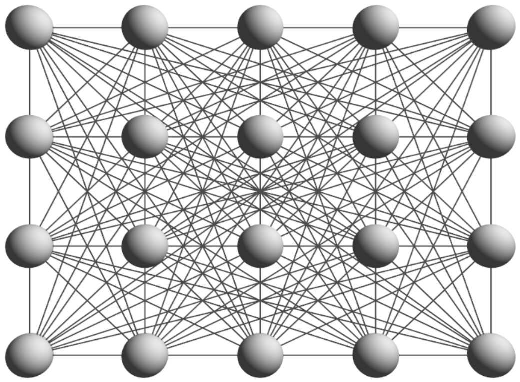
Structured Prediction



Grid CRF

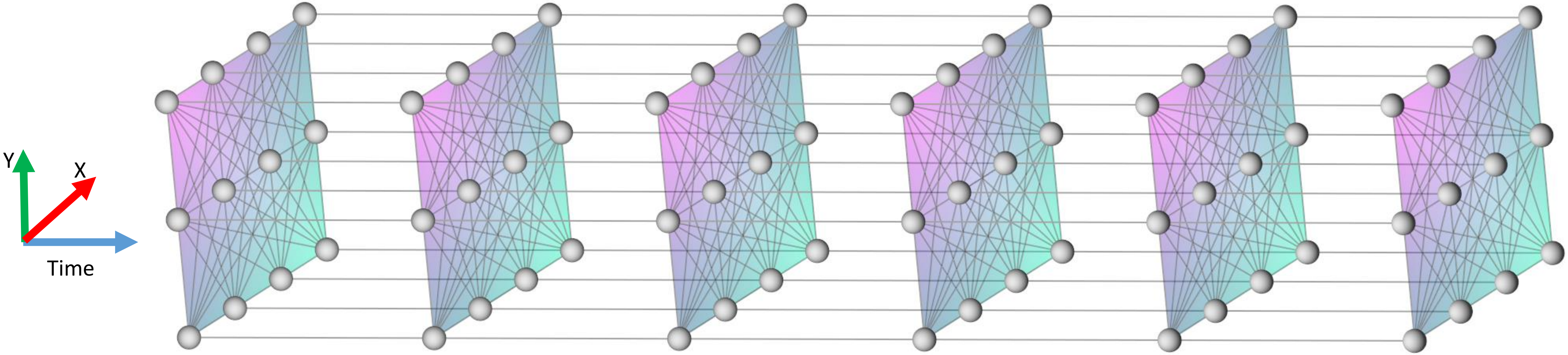


Higher-order CRF

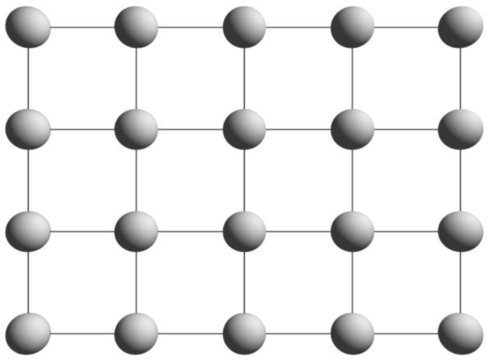


Dense CRF

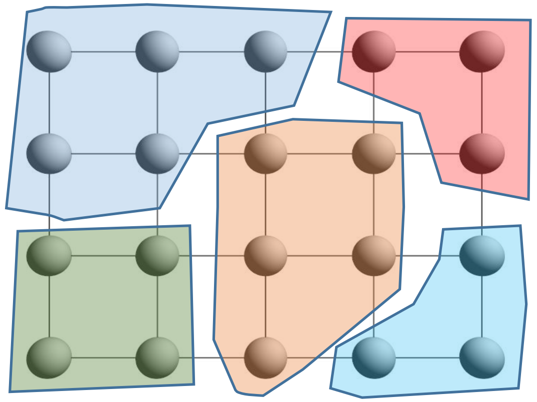
2D dense CRF with sparse temporal edges



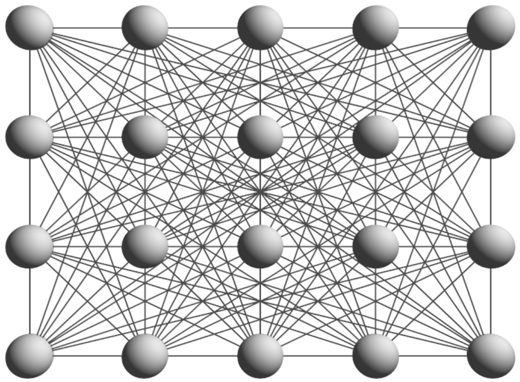
Structured Prediction



Grid CRF

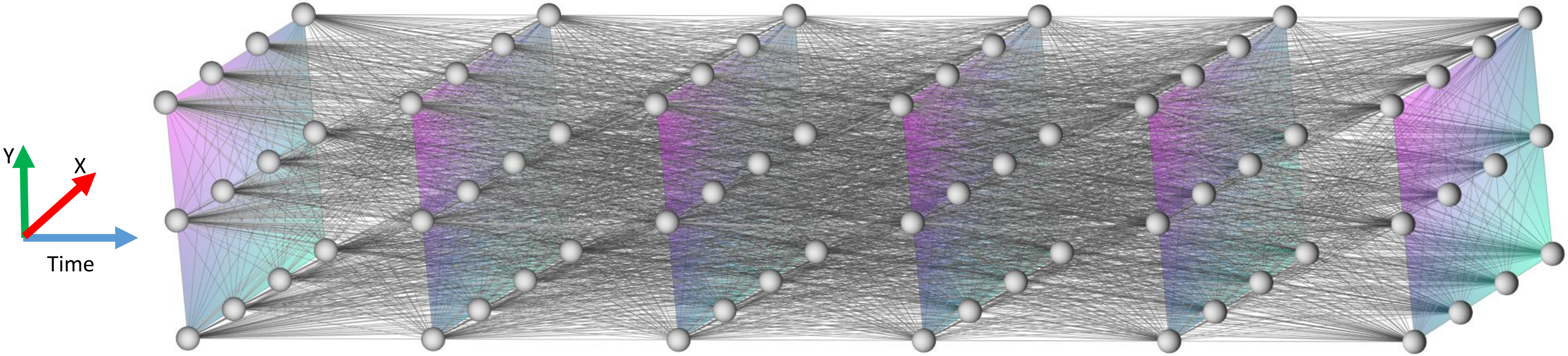


Higher-order CRF



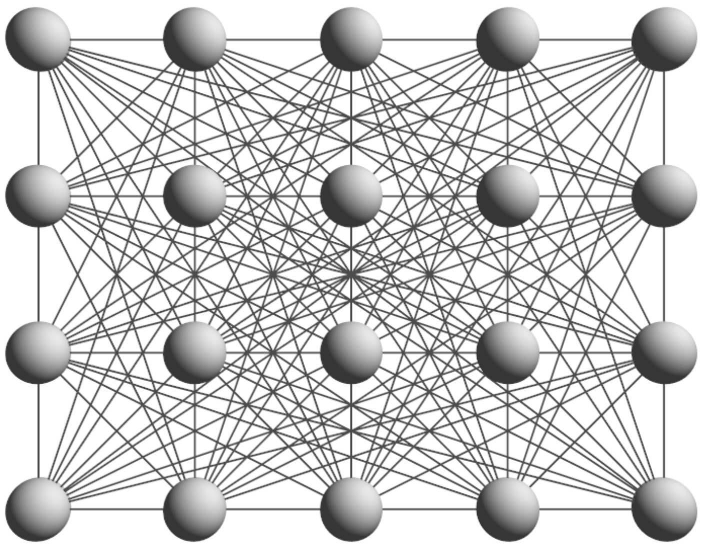
Dense CRF

3D dense CRF



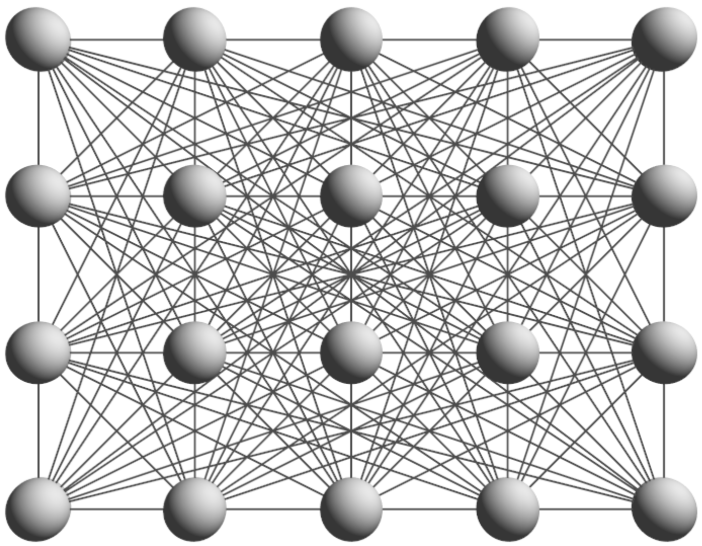
Dense CRF

$$E(x) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j>i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$



Dense CRF

$$E(x) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j>i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$



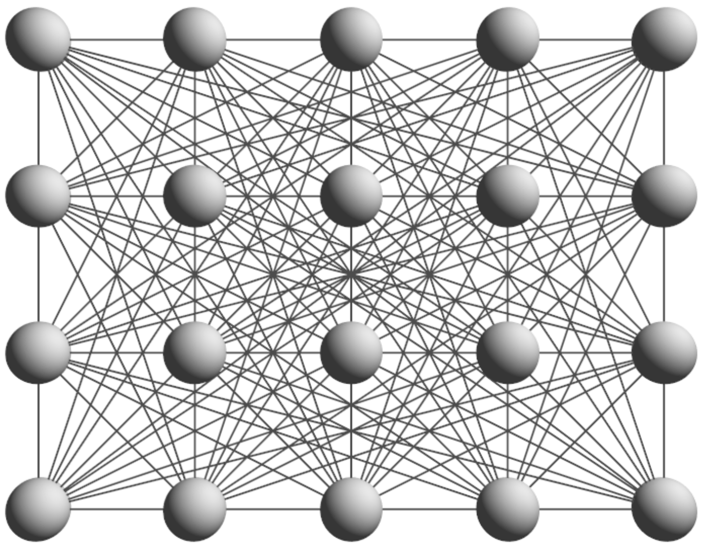
Pairwise potential:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^m \, \kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

Label compatibility function (e.g., Potts function)

Dense CRF

$$E(x) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j>i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$



Pairwise potential:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^m \, \kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

Label compatibility function (e.g., Potts function)

Linear combination of Gaussian kernels

$$\kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j) = \exp\left(-\frac{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|}{\sigma_m^2}\right)$$

$\boldsymbol{f}_i$ is some arbitrary feature space for i^{th} pixel.

Dense CRF

$$E(x) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j>i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$

Pairwise potential:

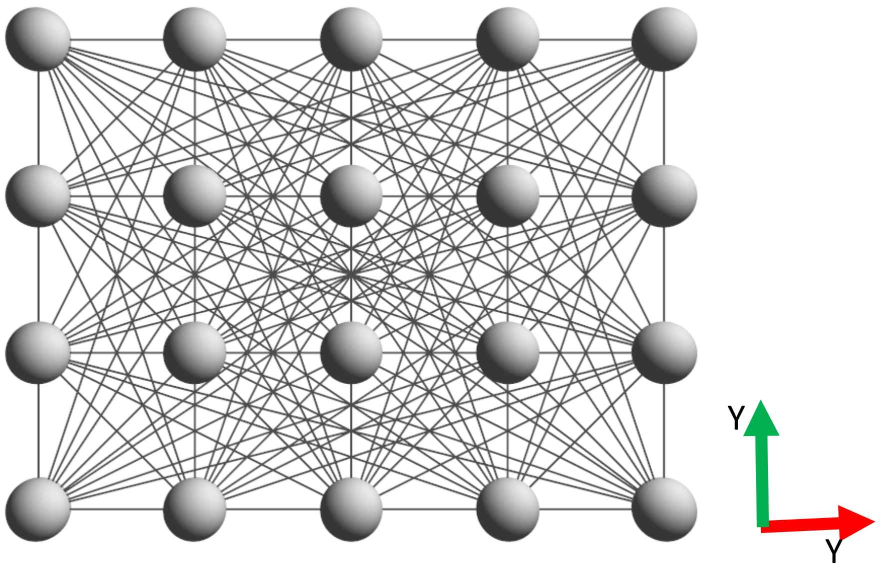
$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^m \kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

Label compatibility function (e.g., Potts function)

Linear combination of Gaussian kernels

$$\kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j) = \exp\left(-\frac{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|}{\sigma_m^2}\right)$$

$\boldsymbol{f}_i$ is some arbitrary feature space for i^{th} pixel.



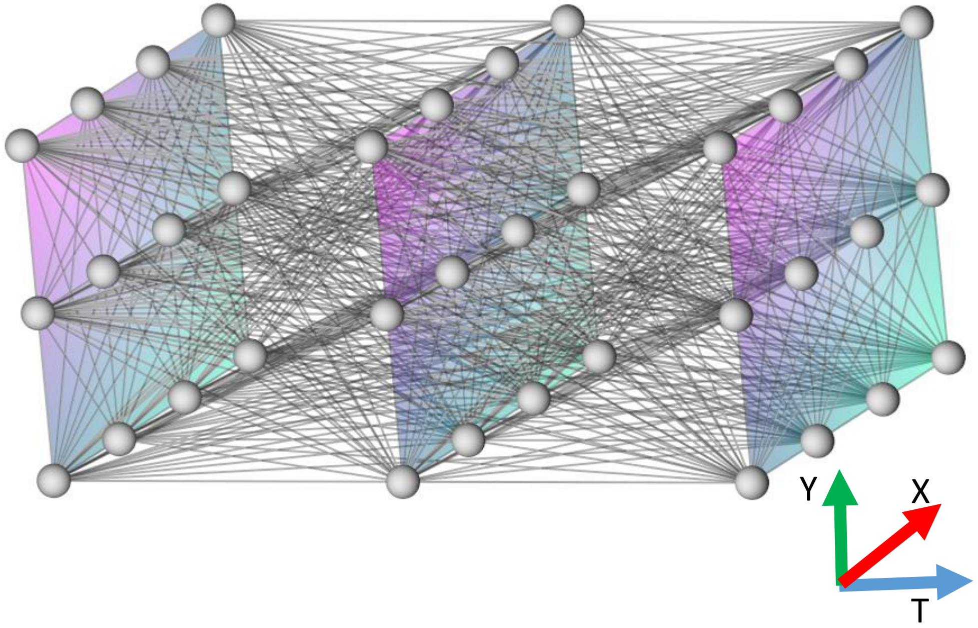
Bilateral feature space

$$\boldsymbol{f}_i = \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix}$$

Provides a contrast-sensitive **spatial** smoothness prior

Dense CRF

$$E(x) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j>i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$



Pairwise potential:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^m \, \kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

Label compatibility function (e.g., Potts function)

Linear combination of Gaussian kernels

$$\kappa^m(\boldsymbol{f}_i, \boldsymbol{f}_j) = \exp\left(-\frac{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|}{\sigma_m^2}\right)$$

$\boldsymbol{f}_i$ is some arbitrary feature space for i^{th} pixel.

$$\boldsymbol{f}_i = \begin{bmatrix} x \\ y \\ t \\ r \\ g \\ b \end{bmatrix}$$

Provides a contrast-sensitive **spatio-temporal** smoothness prior

Feature Space Objective

Are the naïve bilateral feature space e.g. $\begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix}$, $\begin{bmatrix} x \\ y \\ t \\ r \\ g \\ b \end{bmatrix}$ **good**?

Feature Space Objective

- Needs to be low-dimensional (typically single-digit)
  - To be practical for efficient inference

- Features for pixels belonging to the same object should be closer compared to features of two pixels belonging to different semantic objects.

- Corresponding pixels should map to points which are close in the feature space.

Feature Space Objective

- Needs to be low-dimensional (typically single-digit)
  - To be practical for efficient inference

- Features for pixels belonging to the same object should be closer compared to features of two pixels belonging to different semantic objects.

- Corresponding pixels should map to points which are close in the feature space.

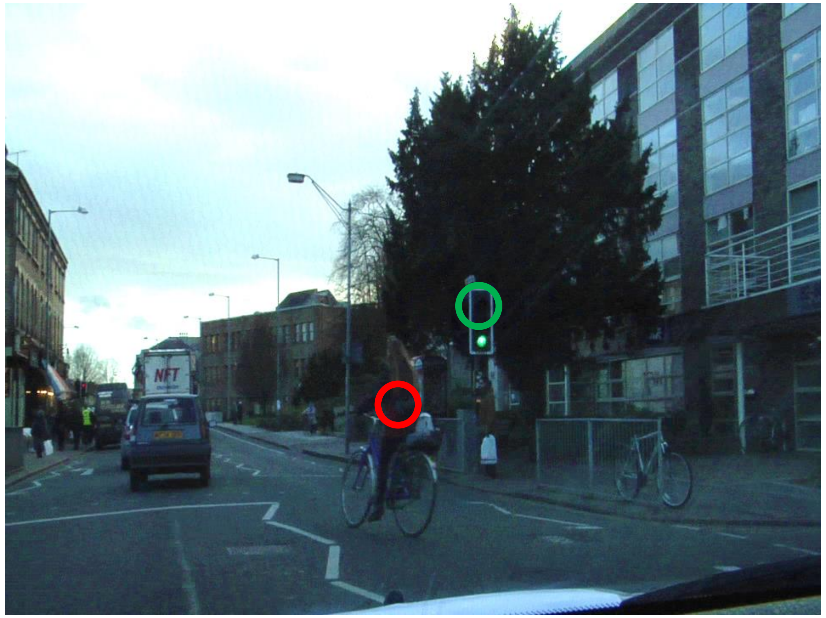
Feature Space Objective

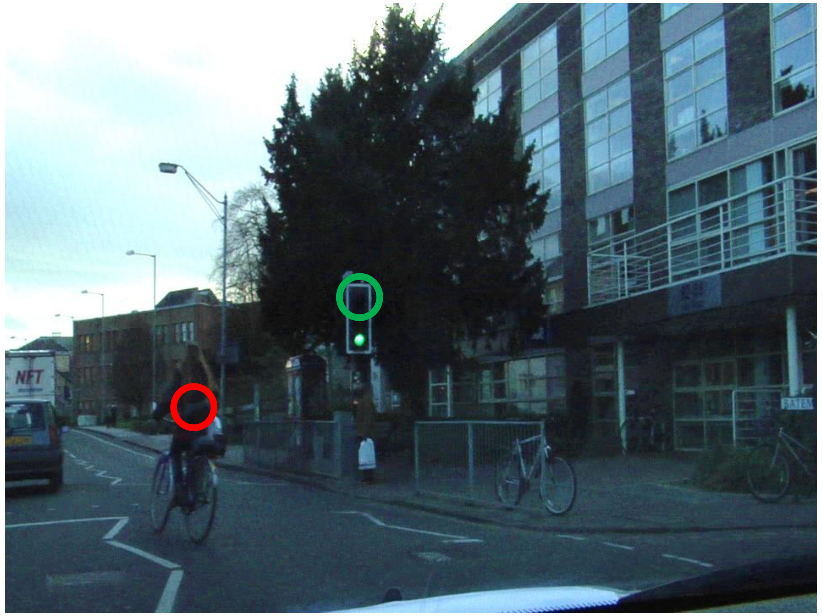
- Needs to be low-dimensional (typically single-digit)
  - To be practical for efficient inference

- Features for pixels belonging to the same object should be closer compared to features of two pixels belonging to different semantic objects.

- Corresponding pixels should map to points which are close in the feature space.

Naïve feature spaces e.g. $\begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix}, \begin{bmatrix} x \\ y \\ t \\ r \\ g \\ b \end{bmatrix}$ does not preserve correspondence





Feature Space Objective

- Needs to be low-dimensional (typically single-digit)
    - To be practical for efficient inference
- Features for pixels belonging to the same object should be closer compared to features of two pixels belonging to different semantic objects.
- Corresponding pixels should map to points which are close in the feature space.

How do we get such a feature space?

Feature Space Objective

- Needs to be low-dimensional (typically single-digit)
  - To be practical for efficient inference

- Features for pixels belonging to the same object should be closer compared to features of two pixels belonging to different semantic objects.

- Corresponding pixels should map to points which are close in the feature space.

How do we get such a feature space?

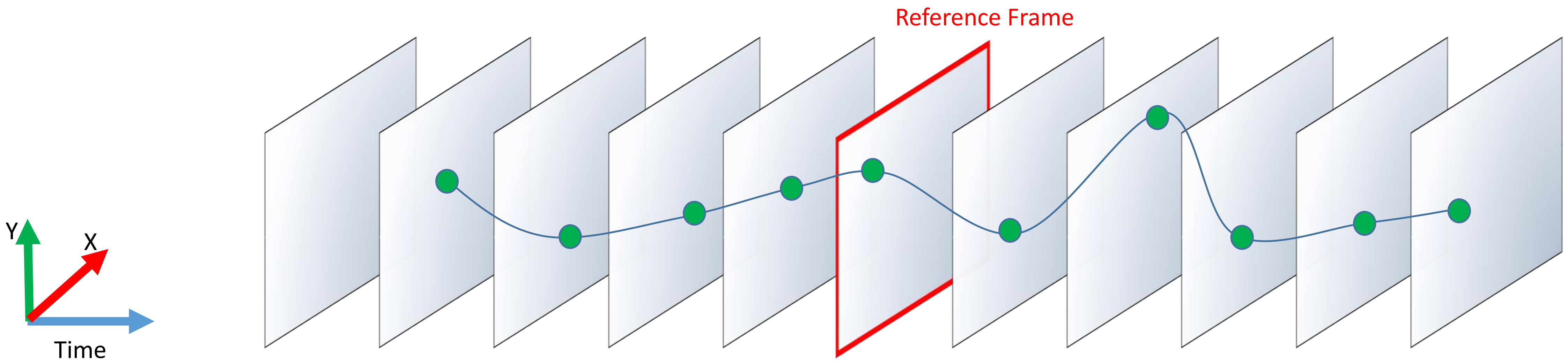
One feature space that satisfies all objectives is the global time-varying 3D coordinate for each pixel. But currently impractical.

Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.

Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.
- Spatio-temporal regularization guided by optical flow and long-term tracks.



Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.
- Spatio-temporal regularization guided by optical flow and long-term tracks.

$$E(\boldsymbol{S}) = \underbrace{E_u(\boldsymbol{S})}_{\text{Data term}} + \underbrace{\gamma_1 E_s(\boldsymbol{S})}_{\text{Spatial smoothness}} + \underbrace{\gamma_2 E_t(\boldsymbol{S})}_{\text{Temporal smoothness}}$$

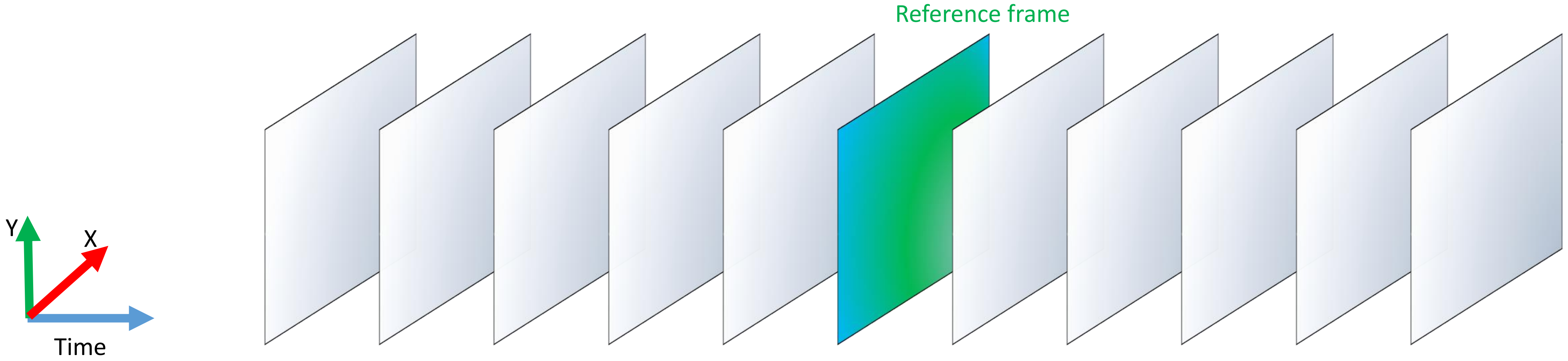
Optimized feature space $\boldsymbol{S}^* = \arg\min_{S} E(\boldsymbol{S})$

Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.
- Spatio-temporal regularization guided by optical flow and long-term tracks.

$$E(\boldsymbol{S}) = \underbrace{E_u(\boldsymbol{S})}_{\text{Data term}} + \underbrace{\gamma_1 E_s(\boldsymbol{S})}_{\text{Spatial smoothness}} + \underbrace{\gamma_2 E_t(\boldsymbol{S})}_{\text{Temporal smoothness}}$$

Optimized feature space $\boldsymbol{S}^* = \arg\min_{S} E(\boldsymbol{S})$

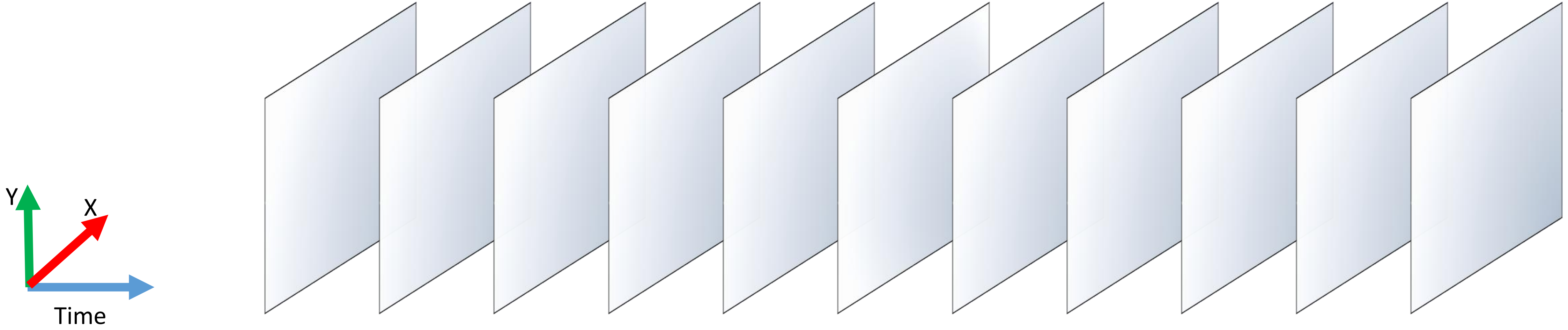


Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.
- Spatio-temporal regularization guided by optical flow and long-term tracks.

$$E(\boldsymbol{S}) = \underbrace{E_u(\boldsymbol{S})}_{\text{Data term}} + \underbrace{\gamma_1 E_s(\boldsymbol{S})}_{\text{Spatial smoothness}} + \underbrace{\gamma_2 E_t(\boldsymbol{S})}_{\text{Temporal smoothness}}$$

Optimized feature space $\boldsymbol{S}^* = \arg\min_{S} E(\boldsymbol{S})$

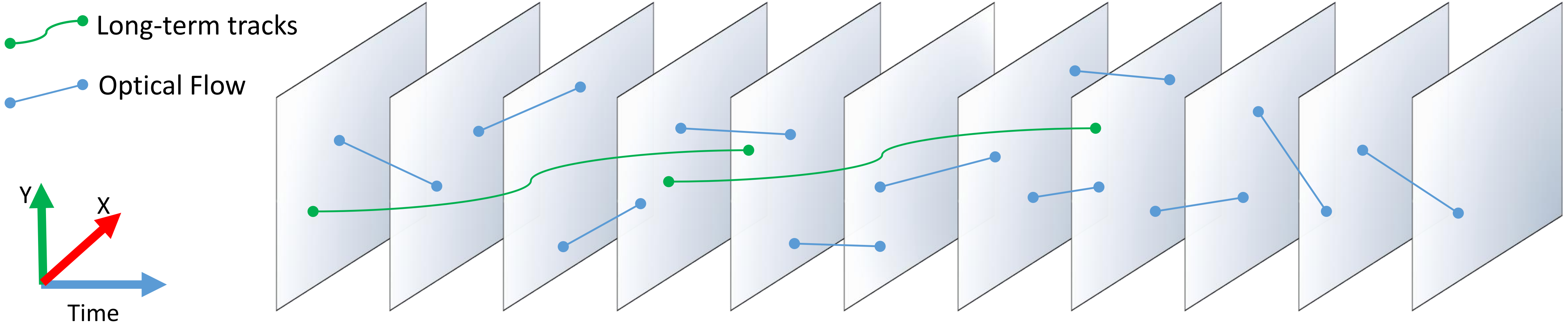


Feature Space Optimization

- Find a feature embedding by optimization that preserves pixel correspondence.
- Spatio-temporal regularization guided by optical flow and long-term tracks.

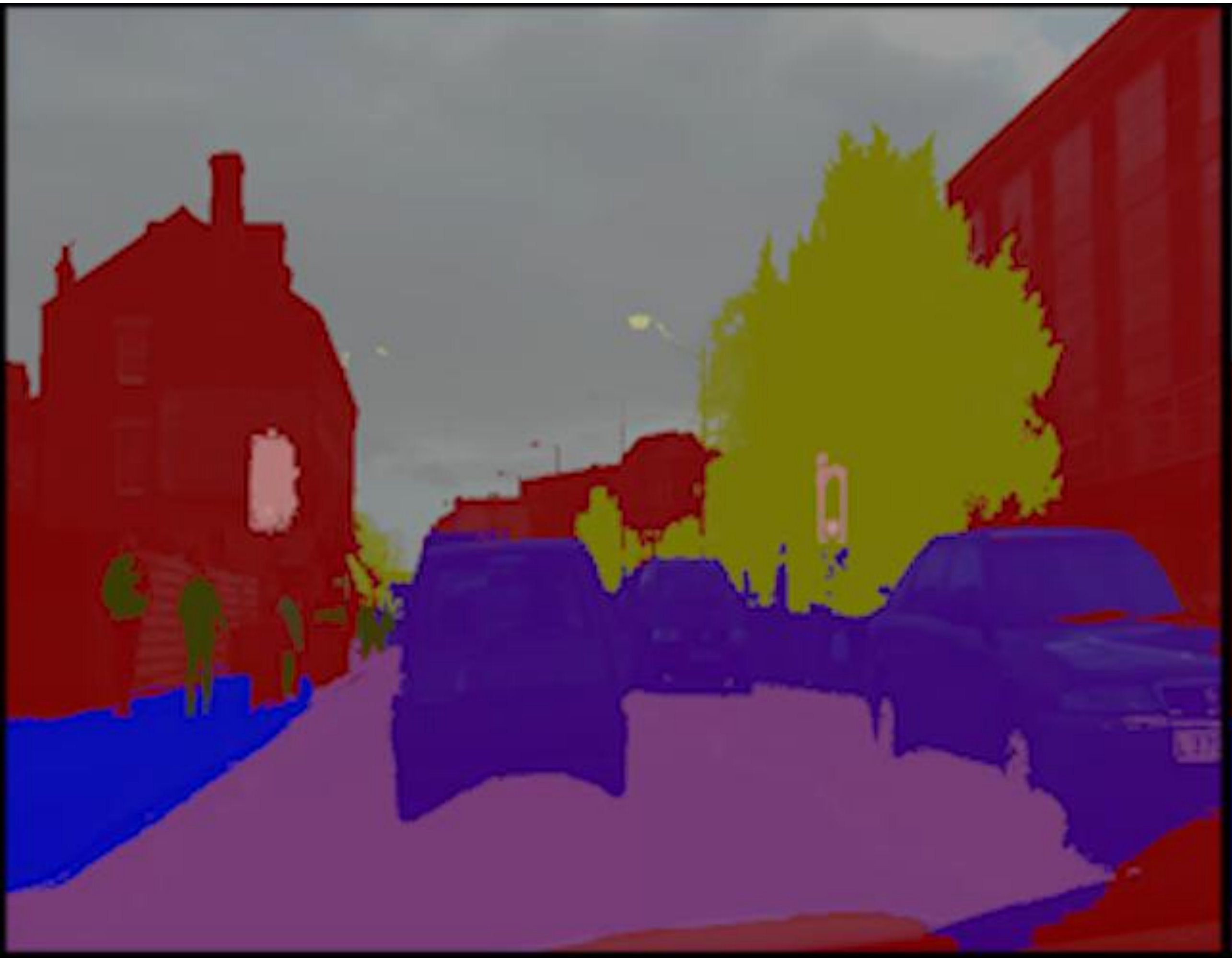
$$E(\boldsymbol{S}) = \underbrace{E_u(\boldsymbol{S})}_{\text{Data term}} + \underbrace{\gamma_1 E_s(\boldsymbol{S})}_{\text{Spatial smoothness}} + \underbrace{\gamma_2 E_t(\boldsymbol{S})}_{\text{Temporal smoothness}}$$

Optimized feature space $\boldsymbol{S}^* = \arg\min_{S} E(\boldsymbol{S})$

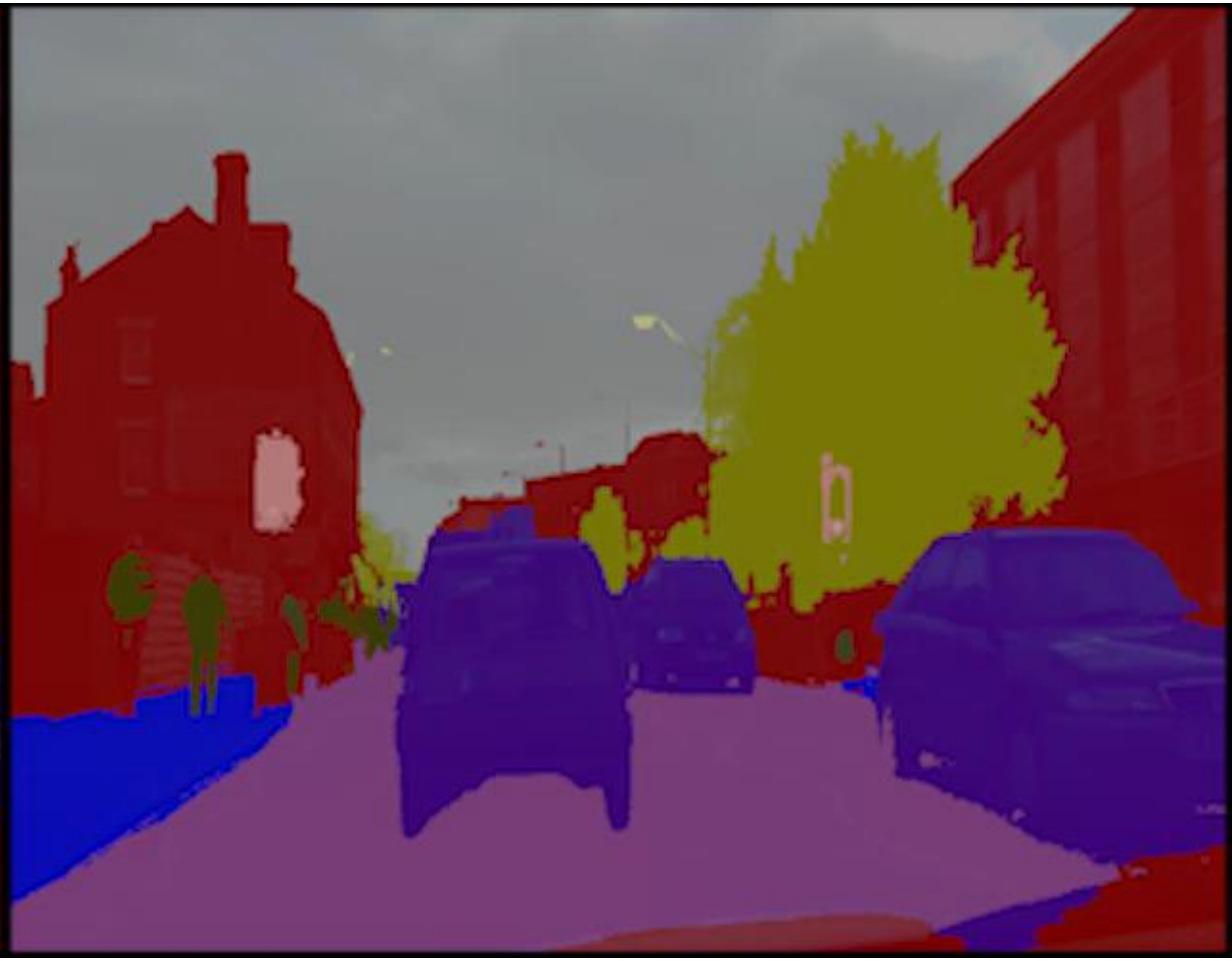


Feature Space Optimization

Without feature optimization



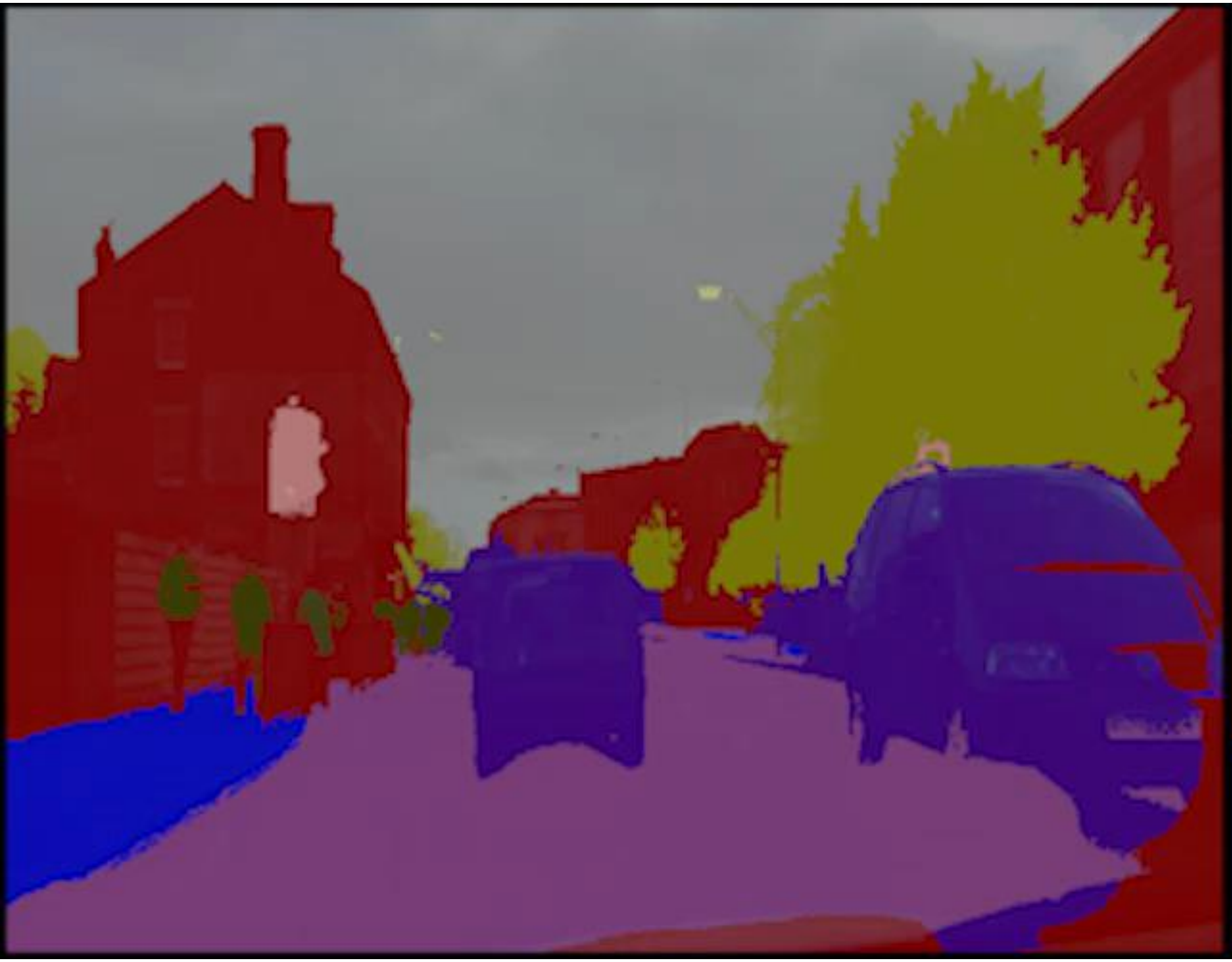
With feature optimization



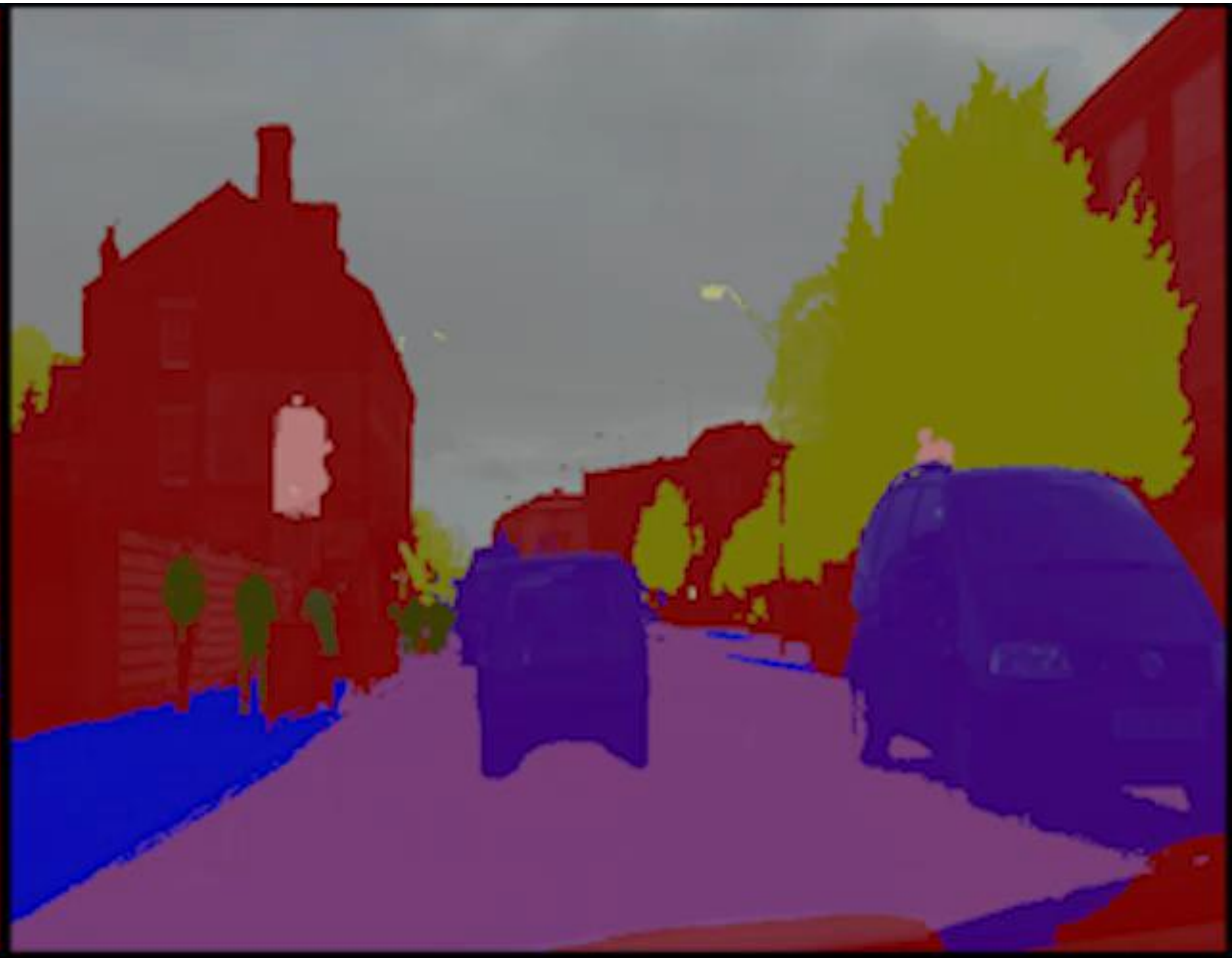
Feature Space Optimization

0.5x (Slow Motion)

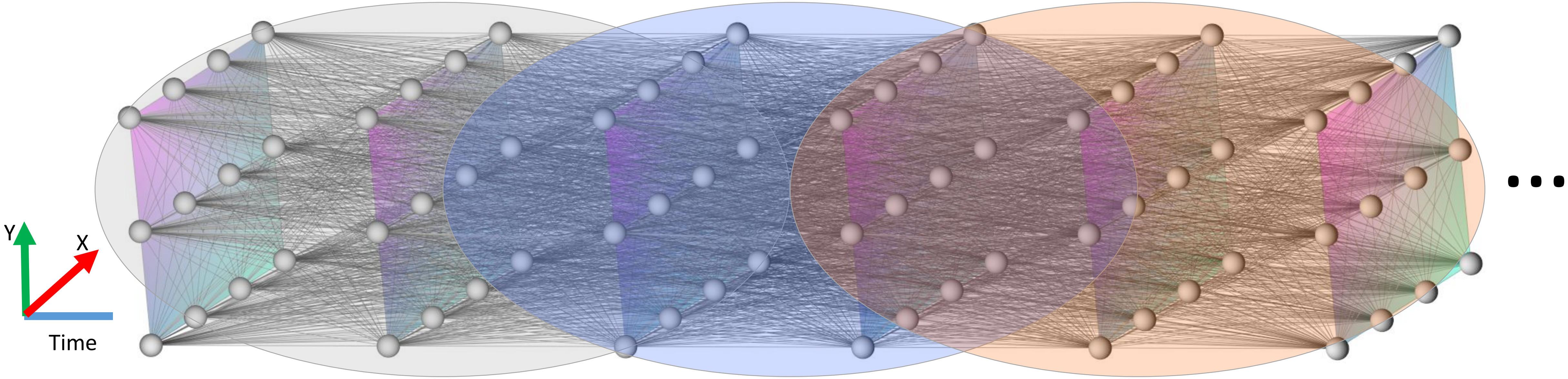
Without feature optimization



With feature optimization



Scaling up to long videos



Overlapping blocks of frames. Each block is a fully-connected dense CRF.

CamVid dataset

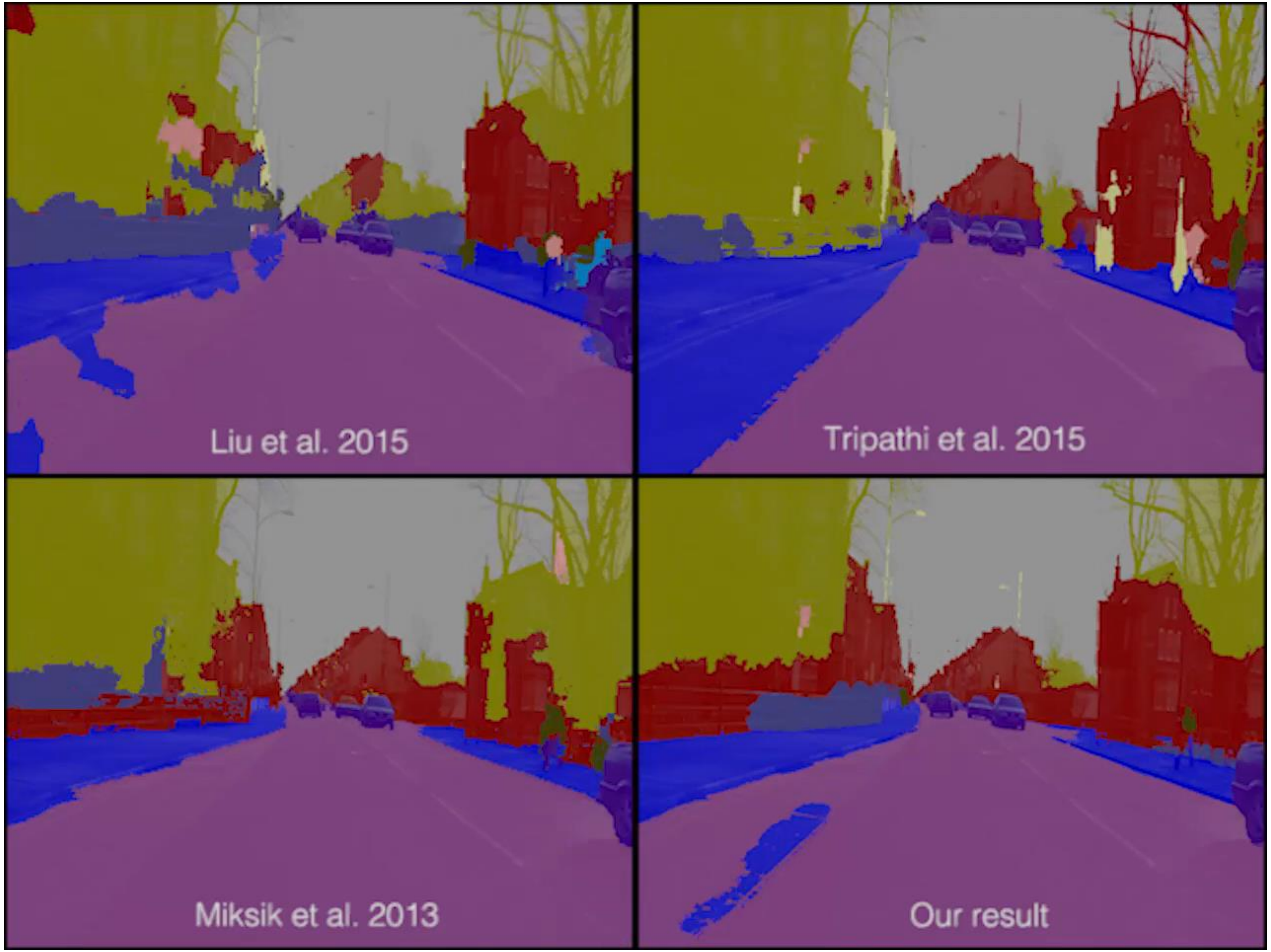
Non-ConvNet unaries	**Mean IOU (%)**	**Temporal consistency (%)**
ALE (Ladicky *et al.* 2009)	53.59	72.2
SuperParsing (Tighe and Lazebnik 2013)	42.03	**88.8**
Tripathi *et al.* 2015	53.18	76.8
Liu and He 2015	47.2	77.6
TextonBoost + Our approach	**55.2**	87.3

ConvNet unaries	**Mean IOU (%)**	**Temporal consistency (%)**
SegNet Basic 2015	46.4	62.5
SegNet Extended 2016	55.6	-
Dilation 2016	65.29	79.0
Dilation + Our approach	**66.12**	**88.3**

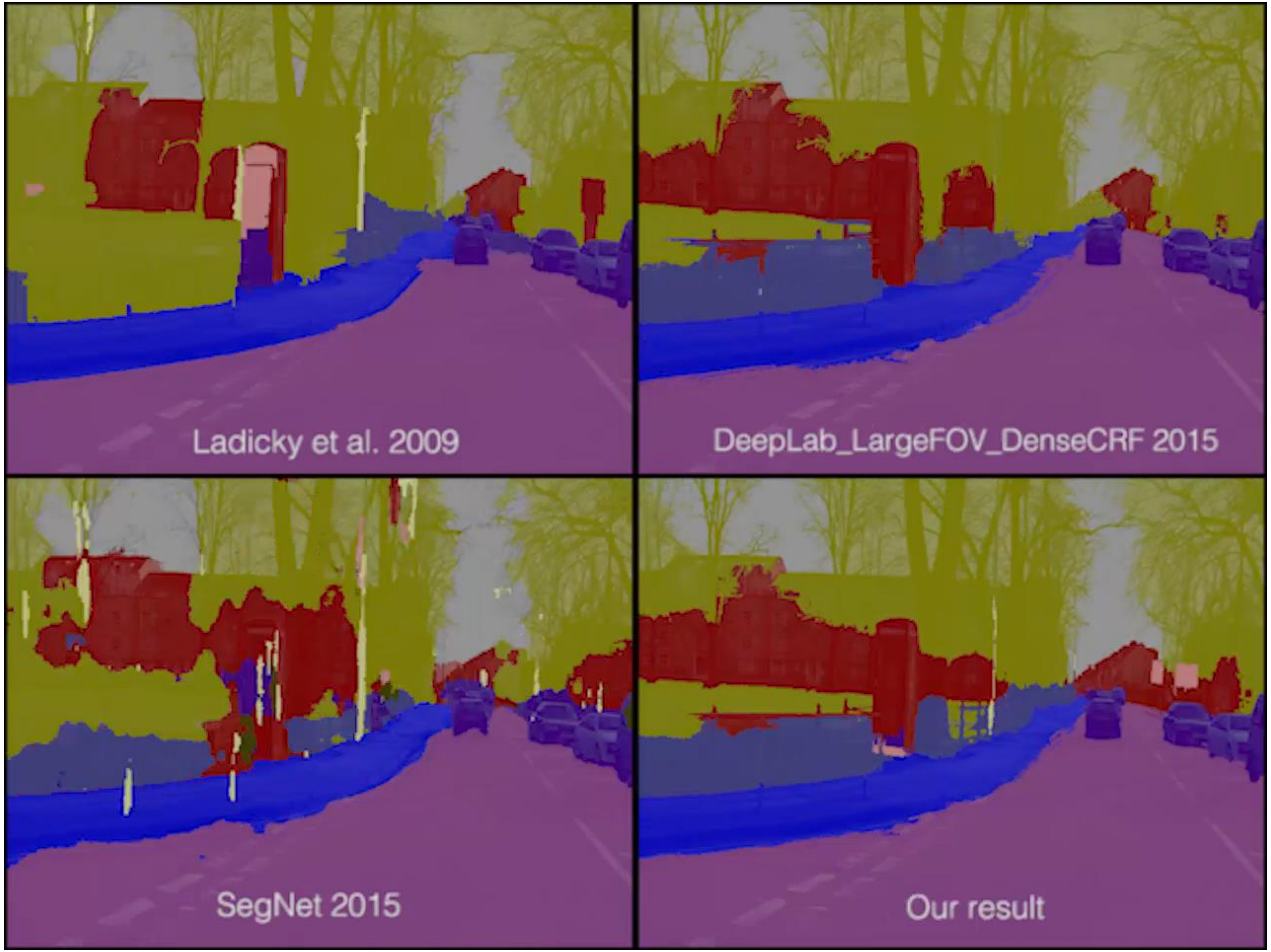
[1]. Brostow *et al*. Semantic object classes in video: A high-definition ground truth database. In PRL, 2009

IOU - intersection over union

Comparison with prior semantic *video* segmentation methods



Comparison with prior semantic image segmentation methods



Cityscapes

- Results on Cityscapes validation set [1]

	Mean IOU (%)	Temporal Consistency (%)
Adelaide [2]	68.6	-
Dilation unary [3]	68.65	88.14
Dilation + Our approach	**70.30**	**94.71**

[1]. Cordts *et al*. The Cityscapes dataset for semantic urban scene understanding. In CVPR, 2016

[2]. Lin *et al*. Efficient piecewise training of deep structured models for semantic segmentation. In CVPR, 2016

[3]. F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In ICLR, 2016

Conclusion

- A CRF model that optimizes over whole video
- Exploits long range context available in video to obtain better and temporally consistent semantic segmentation.
- A low dimensional feature space that captures correspondence information is vital for videos.
- Uses a fast linear solver based optimization to obtain such feature space that captures correspondence information obtained from optical flow.