MITSUBISHI ELECTRIC
Changes for the Better

for a greener tomorrow

Representation Learning Using Adversarially-Contrastive Optimal Transport

Anoop Cherian
MERL, Cambridge, MA

Shuchin Aeron
Tufts University, Medford, MA

MITSUBISHI ELECTRIC
Changes for the Better

MITSUBISHI ELECTRIC
RESEARCH LABORATORIES, INC

Virtual ICML 2020

Tufts UNIVERSITY | School of Engineering

Problem Setup

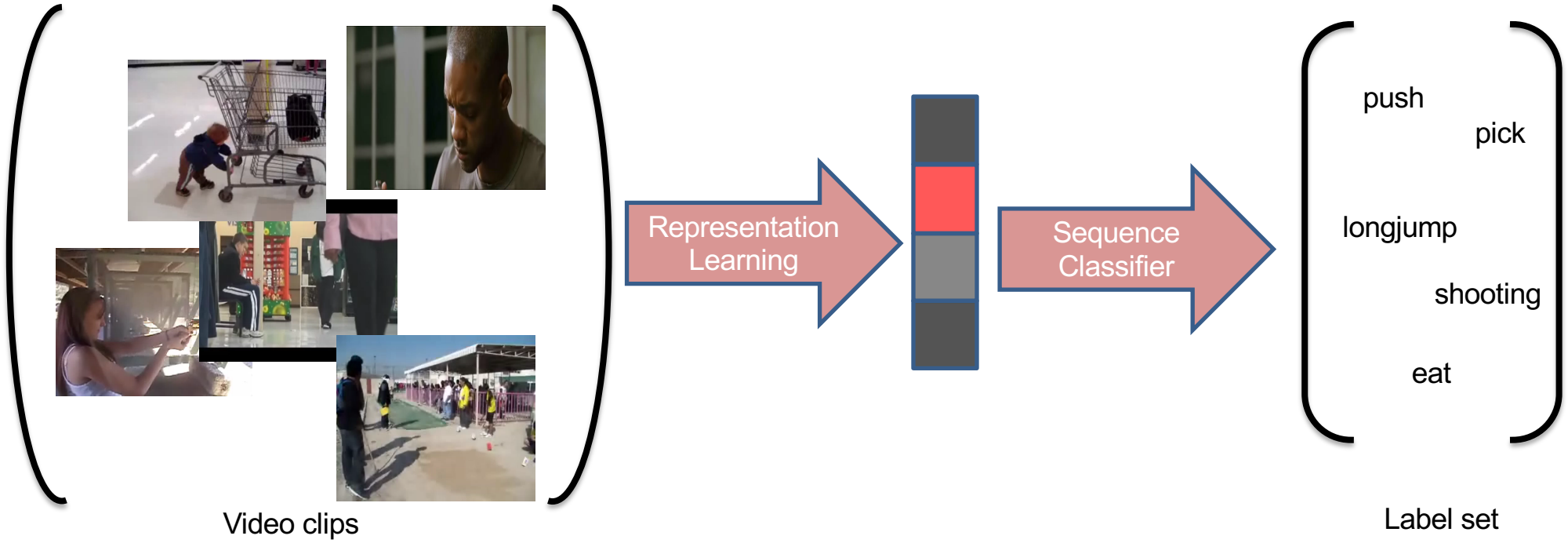
Video clips

Sequence Classifier

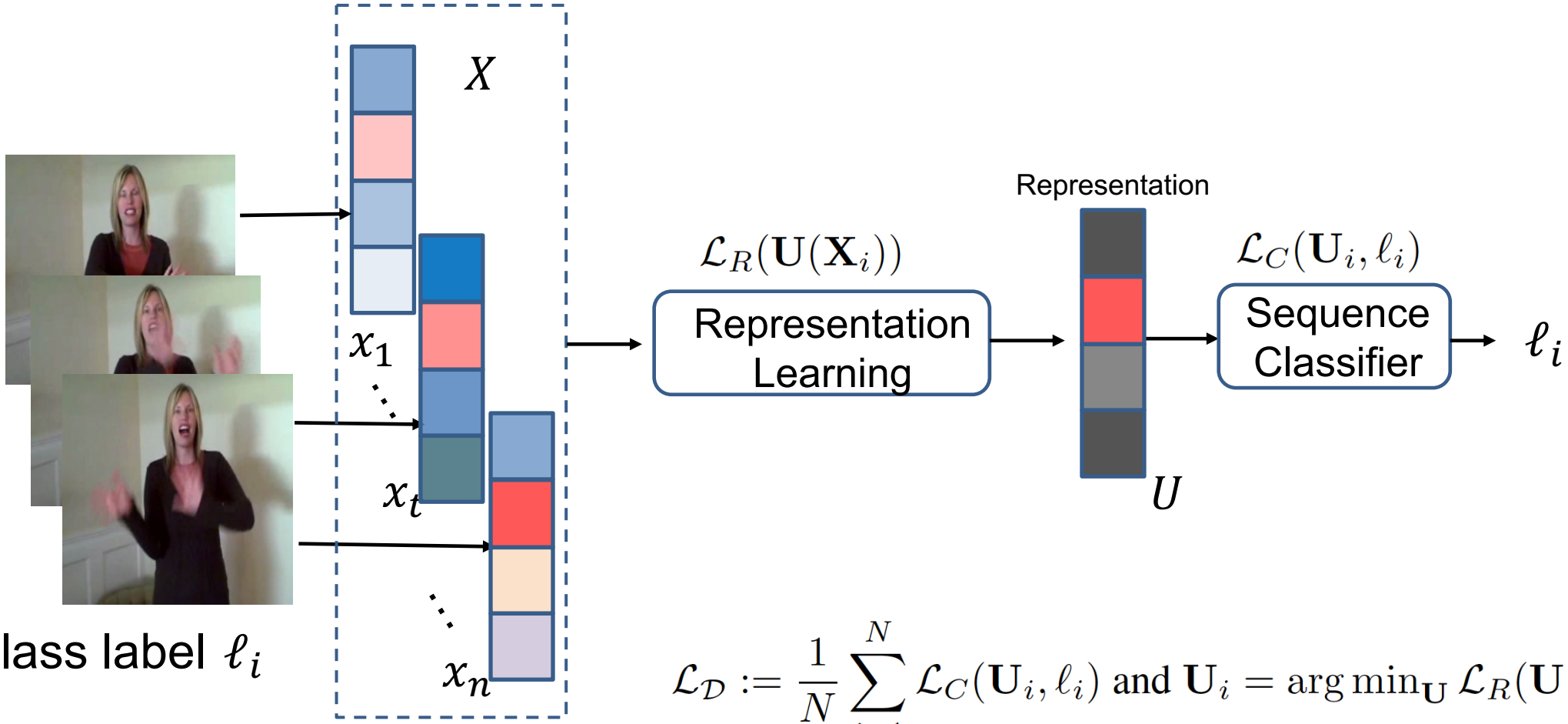
push
pick
longjump
shooting
eat

Label set

Problem Setup

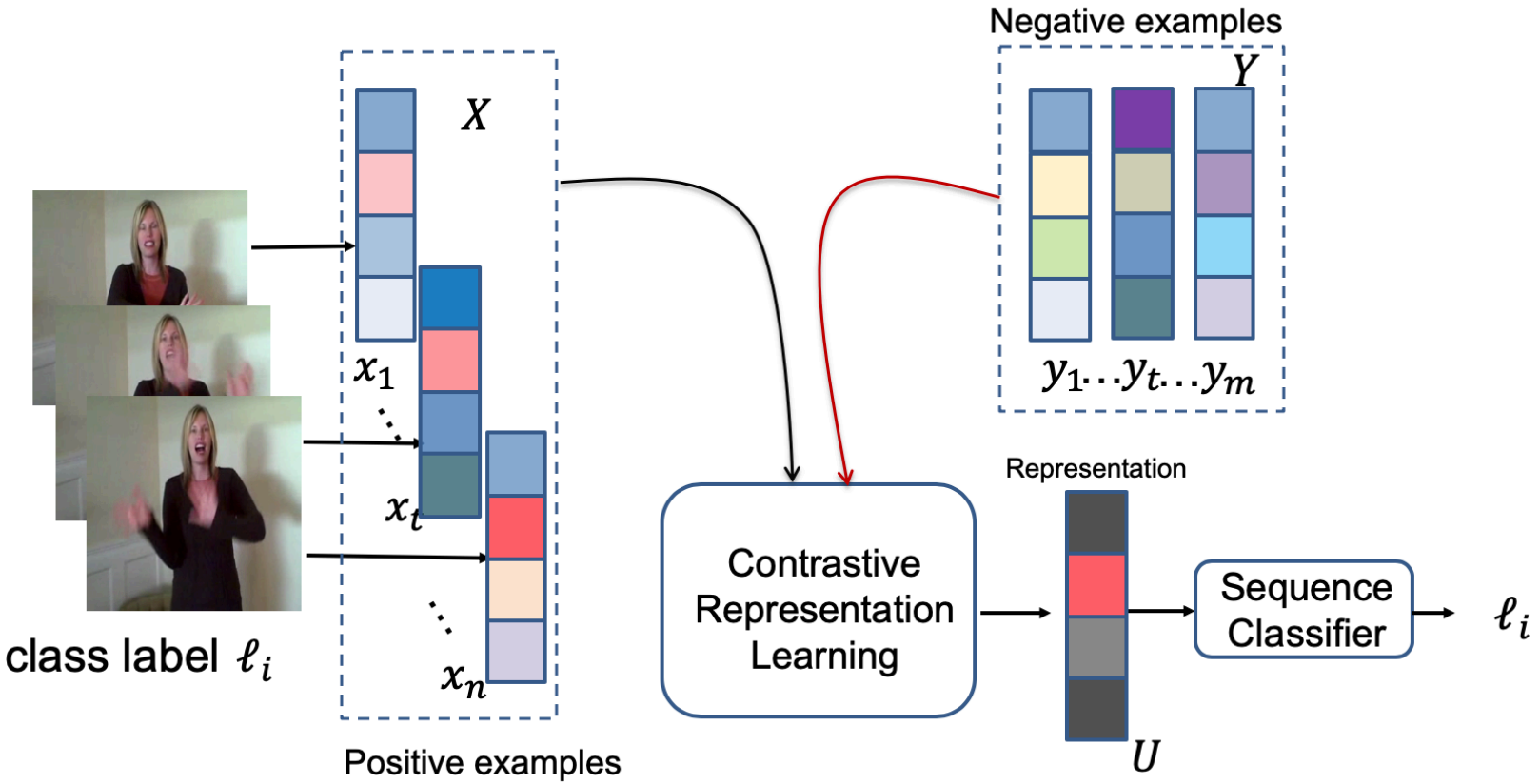


Problem Formulation

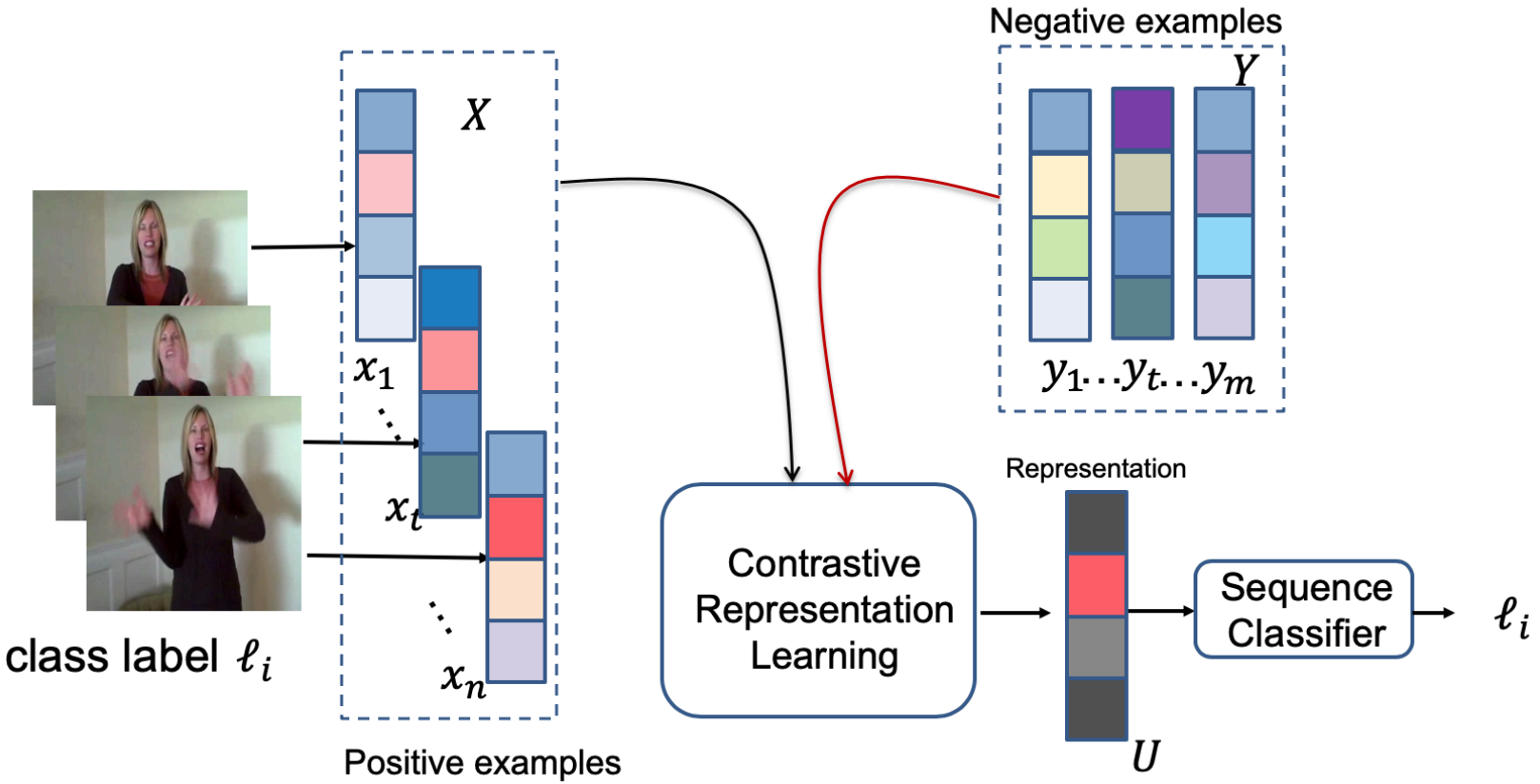


$$\mathcal{L}_{\mathcal{D}} := \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_C(\mathbf{U}_i, \ell_i) \text{ and } \mathbf{U}_i = \arg\min_{\mathbf{U}} \mathcal{L}_R(\mathbf{U}(\mathbf{X}_i))$$

Contrastive Representation Learning

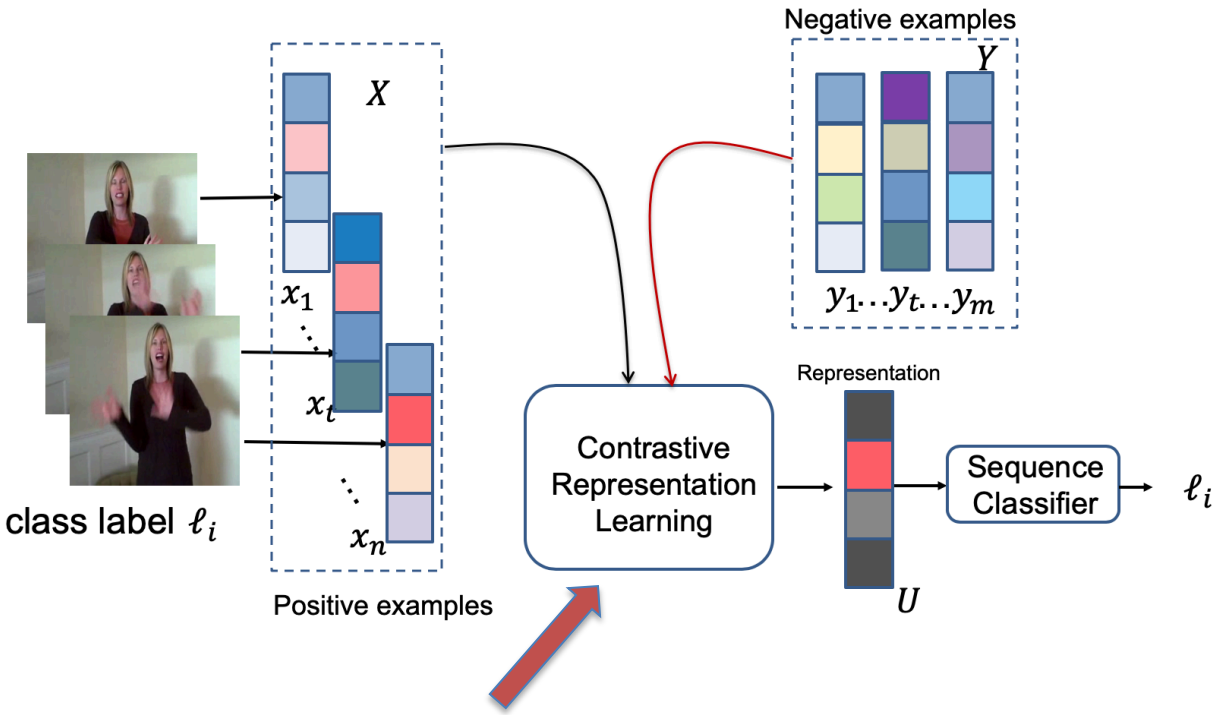


Contrastive Representation Learning



Key idea: The learned representation is as close as possible to X and as far as possible to the negative samples Y.

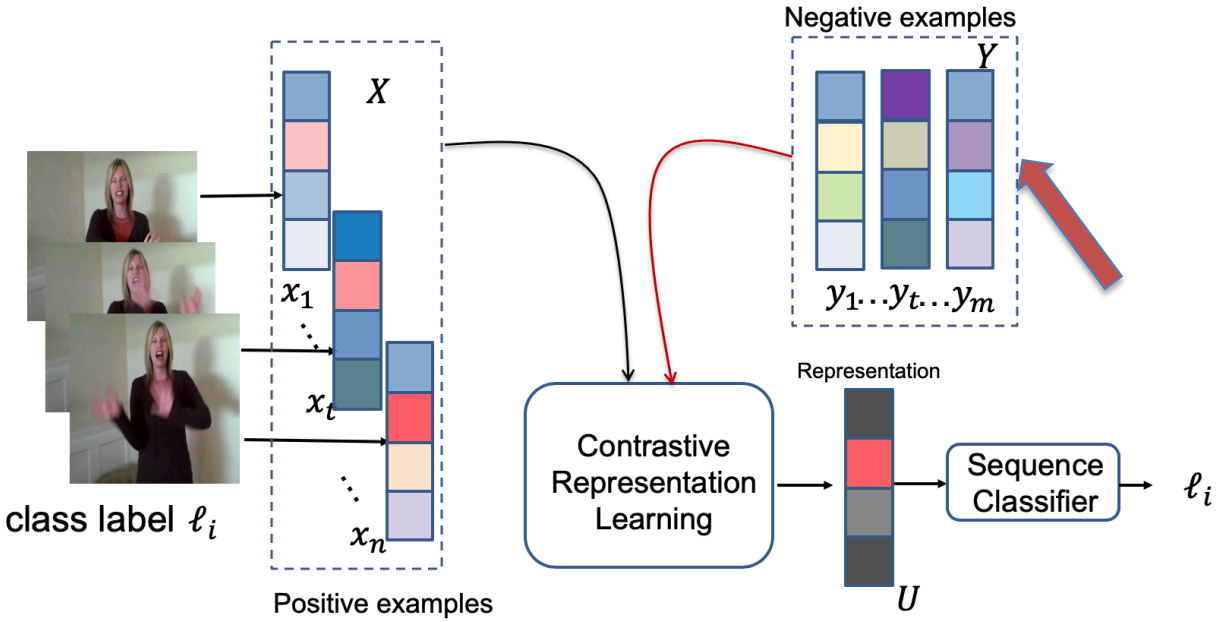
Contrastive Representation Learning



Questions:

1. How should we characterize contrastiveness?

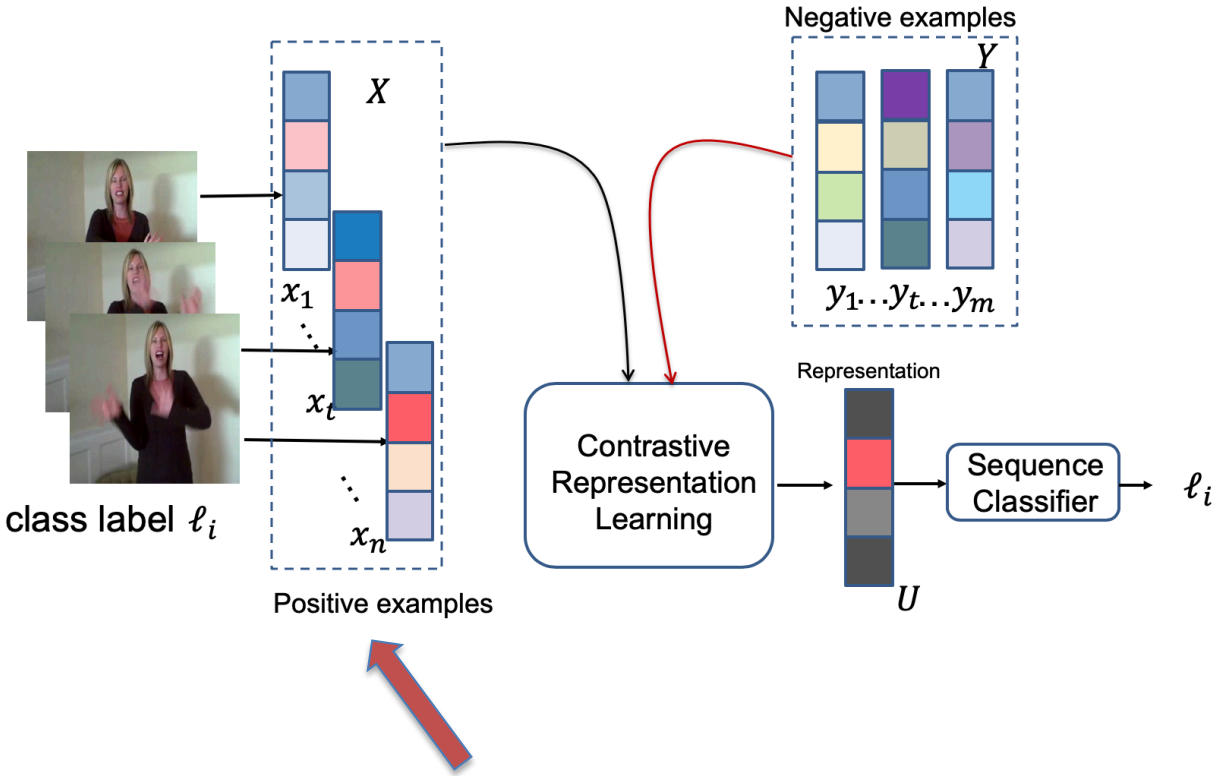
Contrastive Representation Learning



Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?

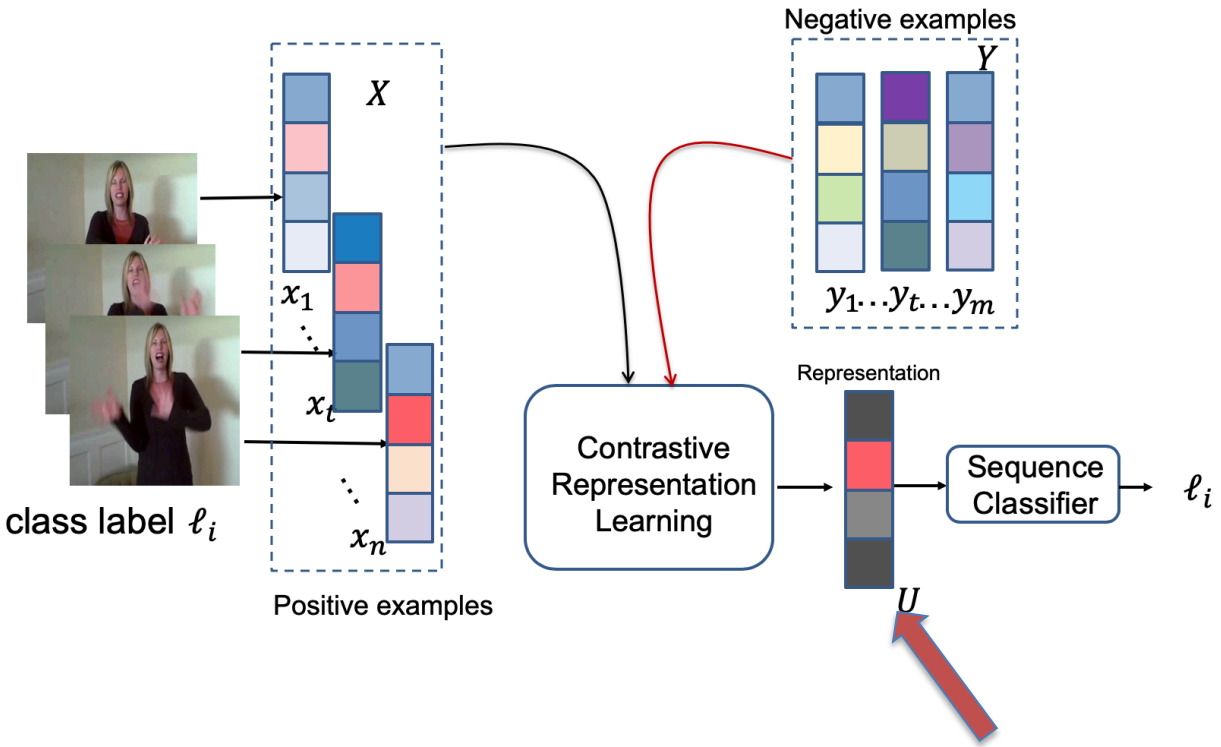
Contrastive Representation Learning



Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?

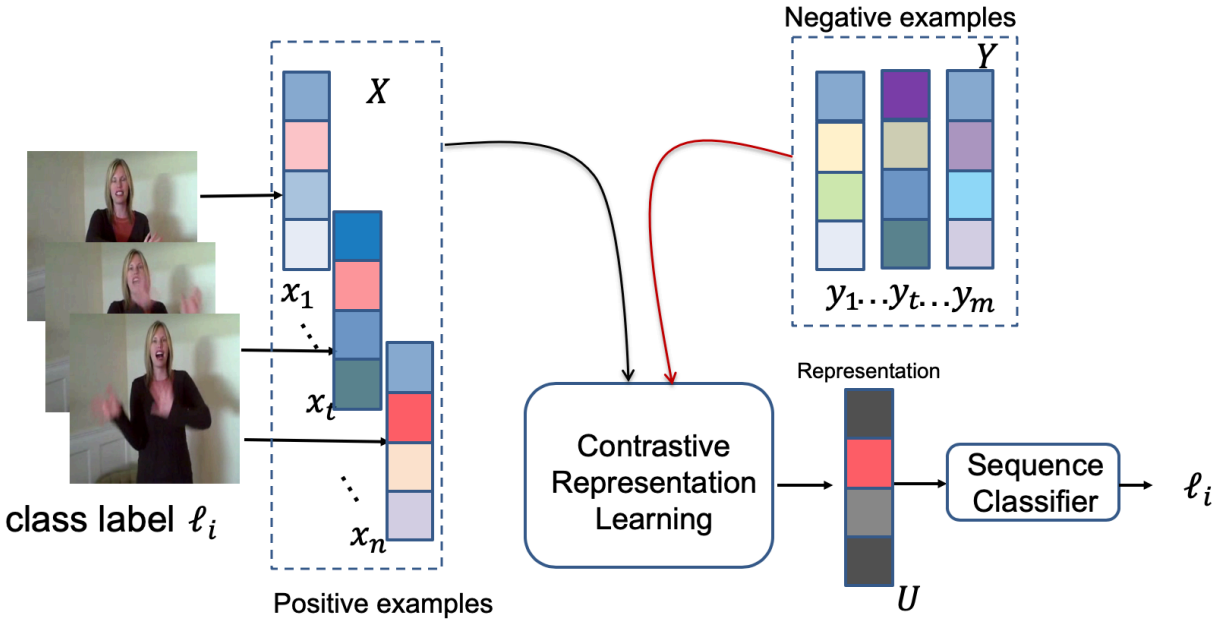
Contrastive Representation Learning



Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?
4. How should we “represent” the sequence?

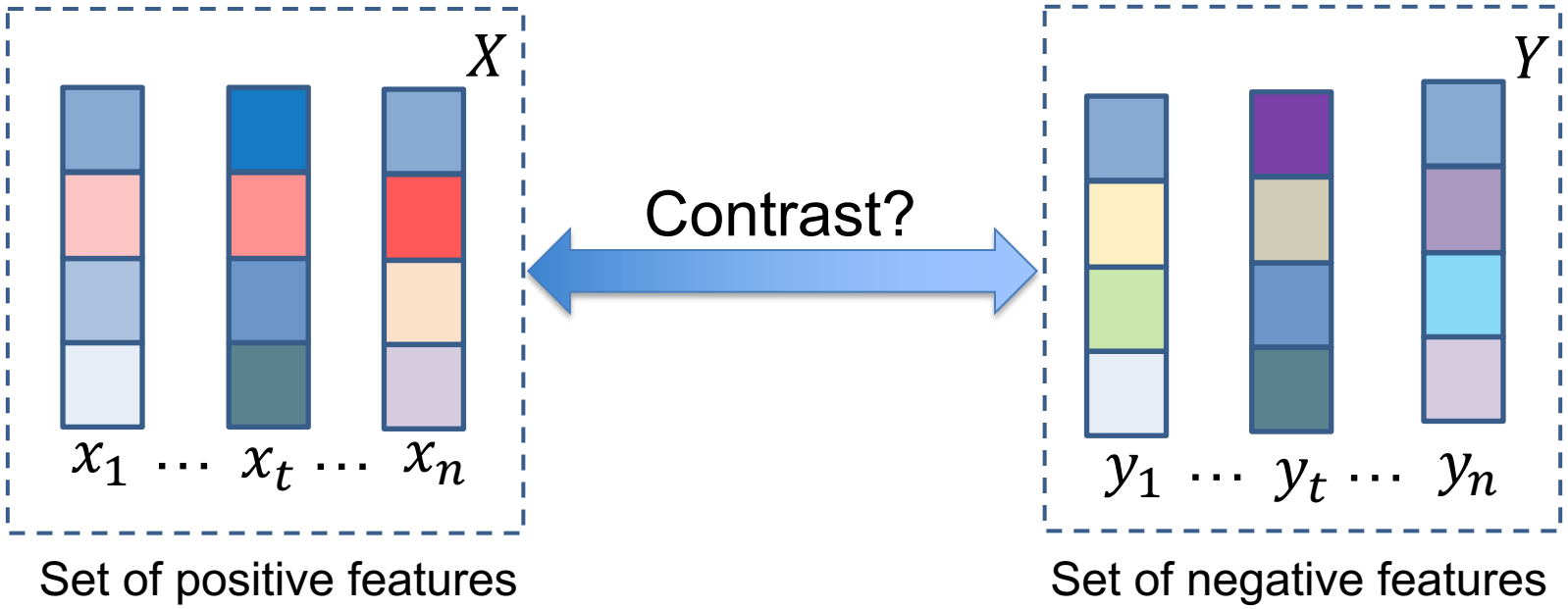
Contrastive Representation Learning



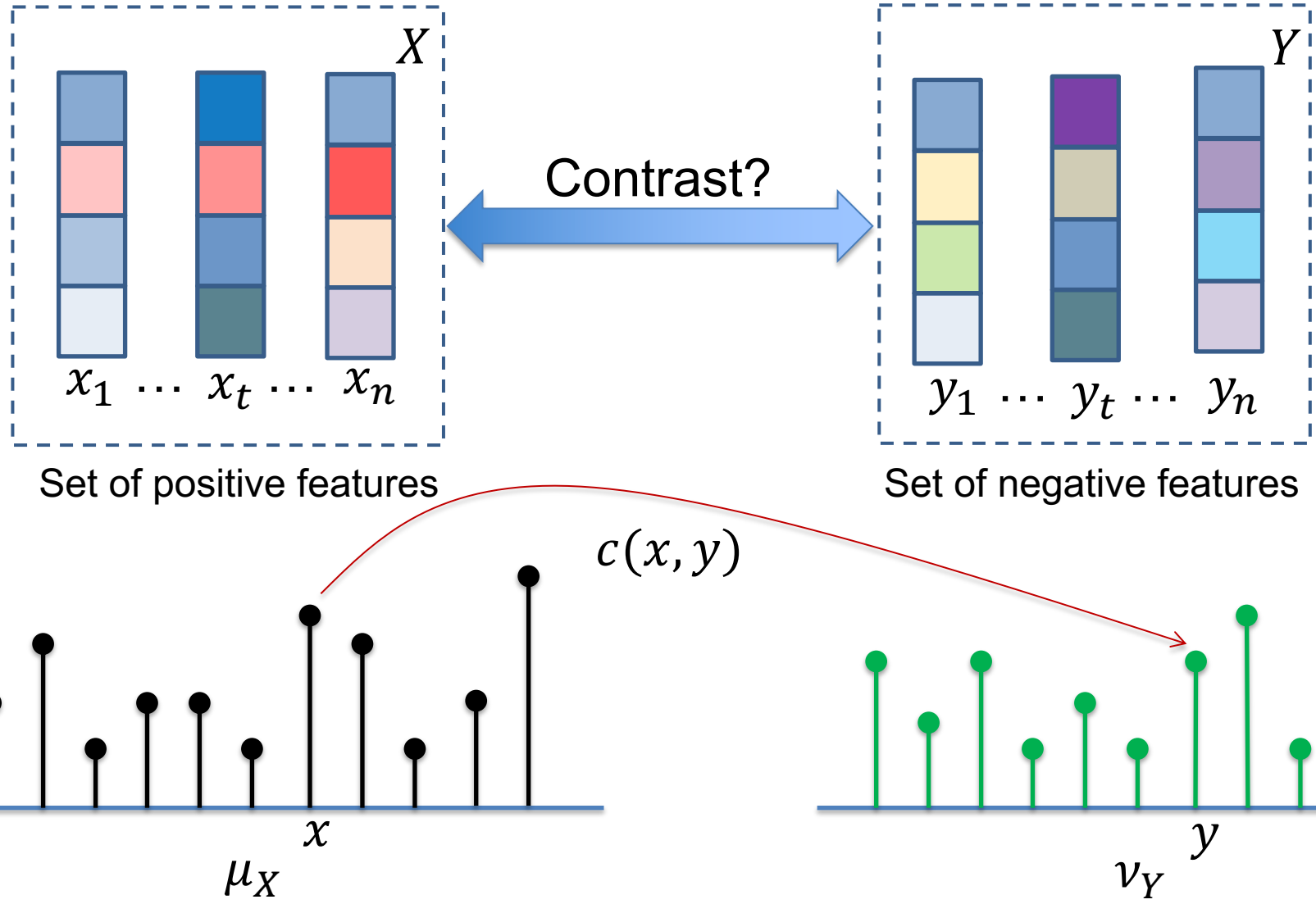
Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?
4. How should we “represent” the sequence?

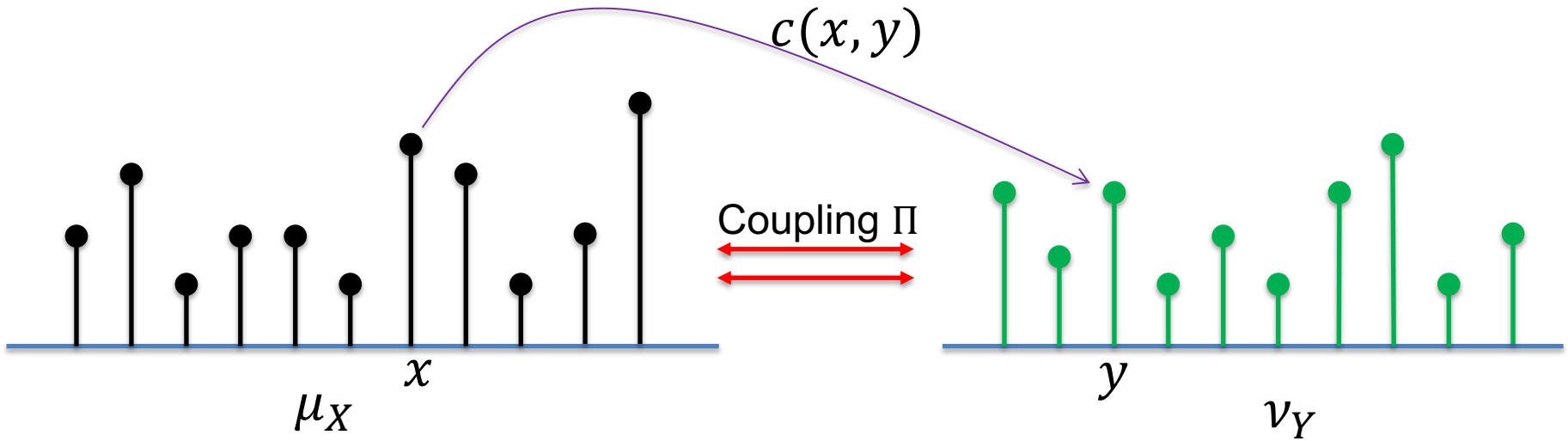
How Should We Characterize Contrastiveness?



How Should We Characterize Contrastiveness?



How Should We Characterize Contrastiveness?

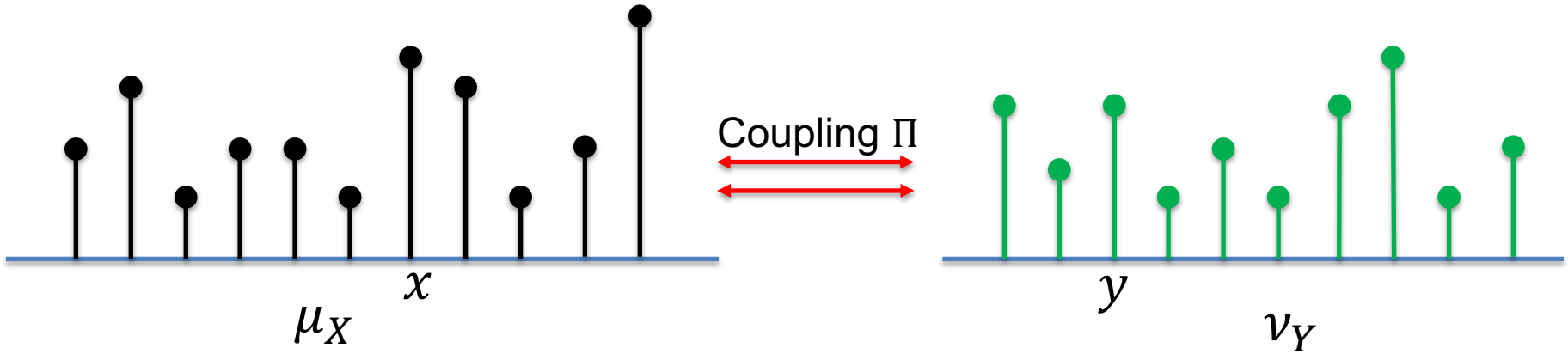


Wasserstein Distance

$$W_c(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\, c(\mathbf{x}, \mathbf{y})$$

where Π is the set of all coupling between μ and ν.

How Should We Characterize Contrastiveness?

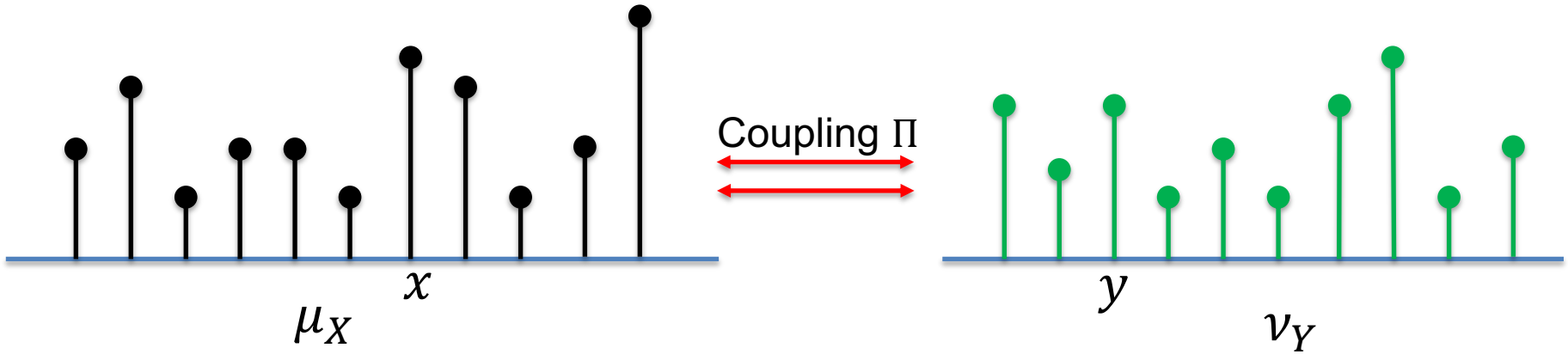


Key Idea: We propose to characterize "contrastiveness" as finding a representation U that maximizes the Wasserstein Distance

$$\max_{\mathbf{U}} \mathcal{L}_{OT}(\mathbf{U}) := W_c(f_{\mathbf{U}\#}\mu_{\mathbf{X}}, \nu_{\mathbf{Y}})$$

Similar to Subspace Robust Optimal Transport [1], we use $f_U = UU^T$, the projection operator for Grassmannian $U \in \mathcal{G}(d,k), U^TU = I_k$.

Contrastive Learning via Optimal Transport

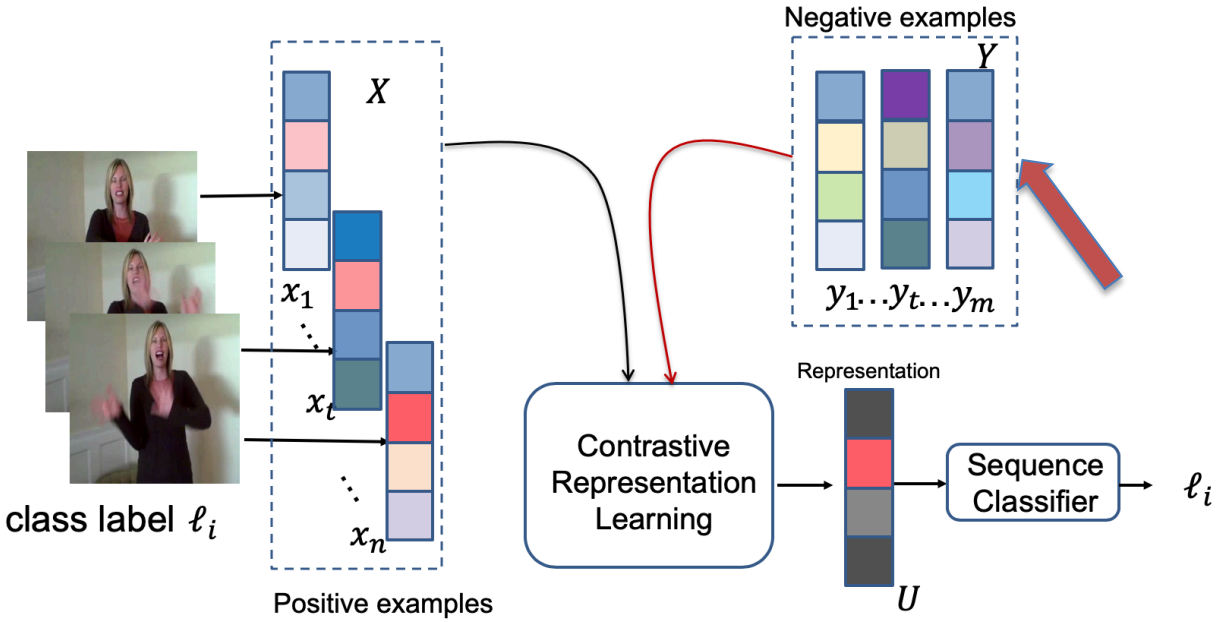


$$\max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathcal{L}_{OT}(\mathbf{U}) := \inf_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \sum_{i,j} \pi_{ij} \| f_{\mathbf{U}}(\mathbf{x}_i) - \mathbf{y}_j \|$$

What does this mean?

It asks to find a subspace $\mathbf{U}$ on a Grassmann manifold such that projections of $x_i \in X$ onto $\mathbf{U}$ will maximize their Wasserstein distance from the negatives, $y_j \in Y$. Such a subspace $\mathbf{U}$ should thus capture discriminative properties between X and Y.

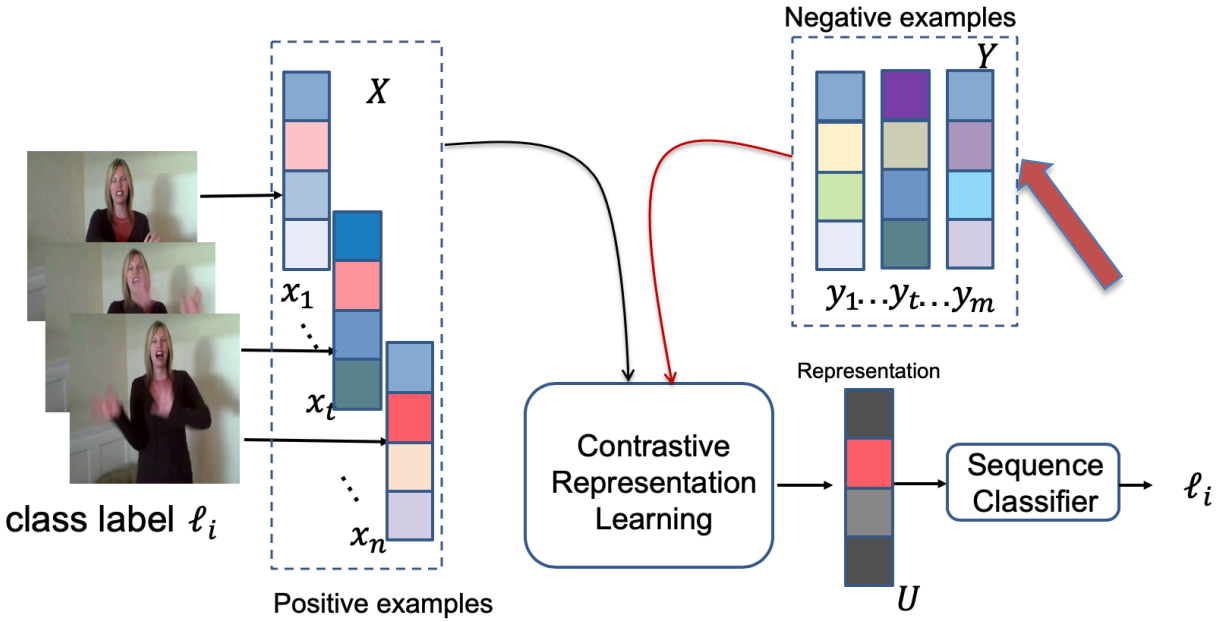
Contrastive Representation Learning



Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?
4. How should we "represent" the sequence?

Negative Examples



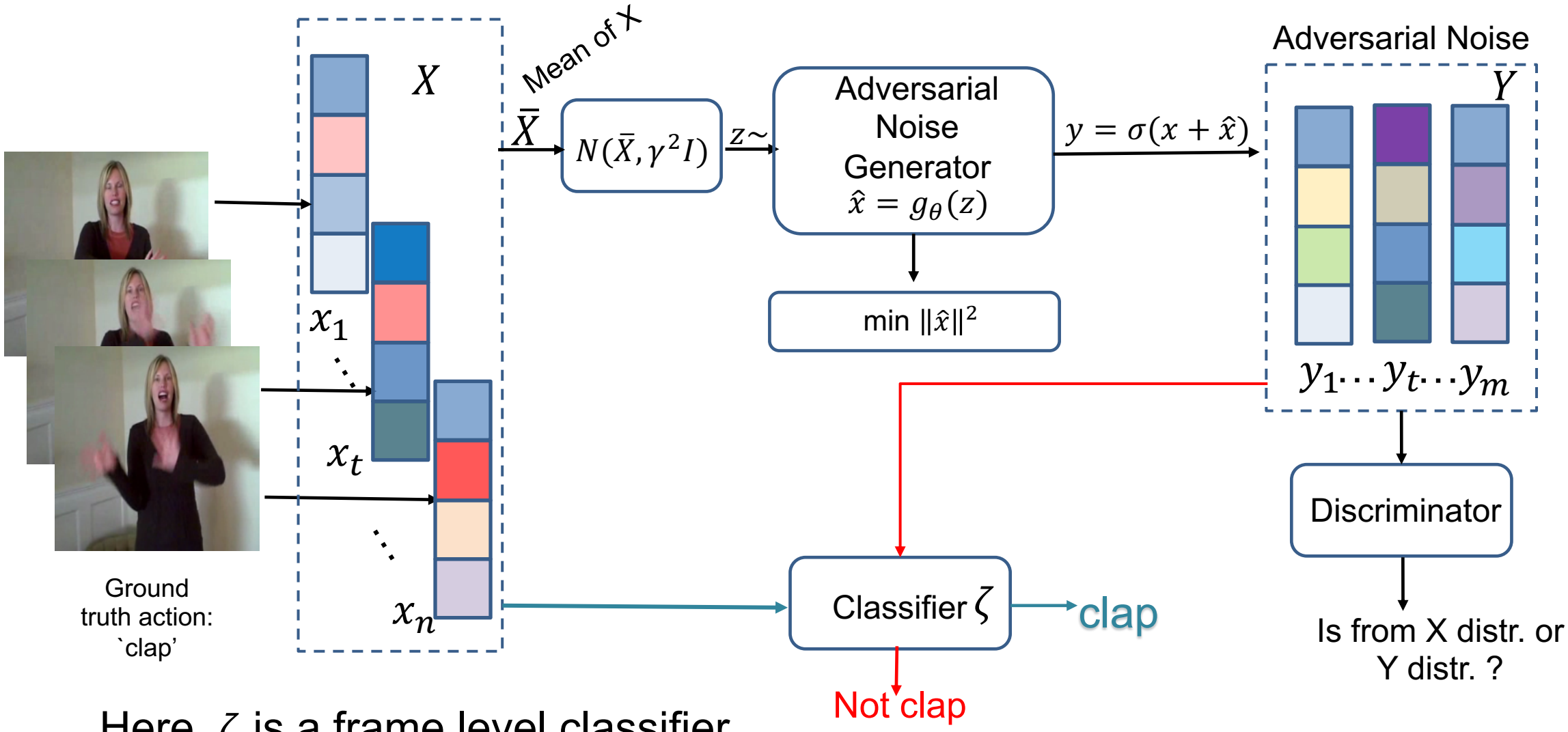
Questions:

a) How can we find negative examples?

b) How can we ensure they do not have useful features?

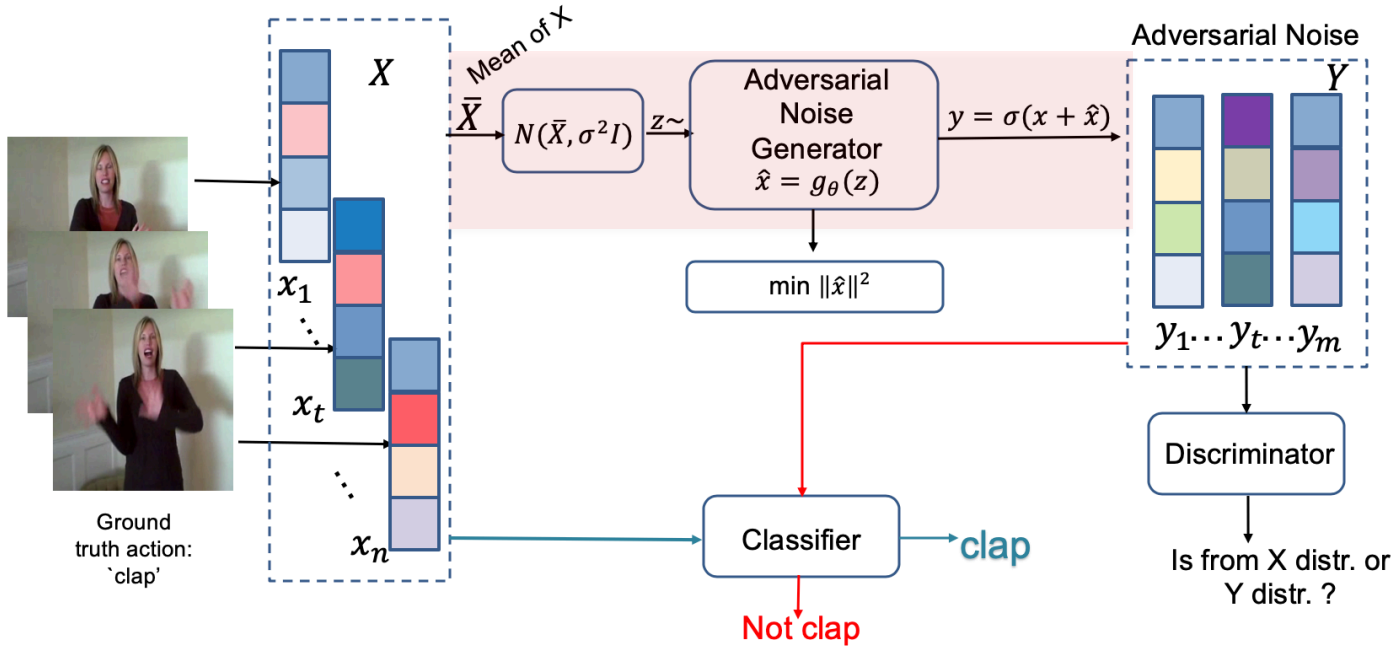
We propose to train an adversarial generator to produce negative feature distribution

Adversarial Distribution Learning Using Wasserstein GAN



Here, ζ is a frame level classifier.

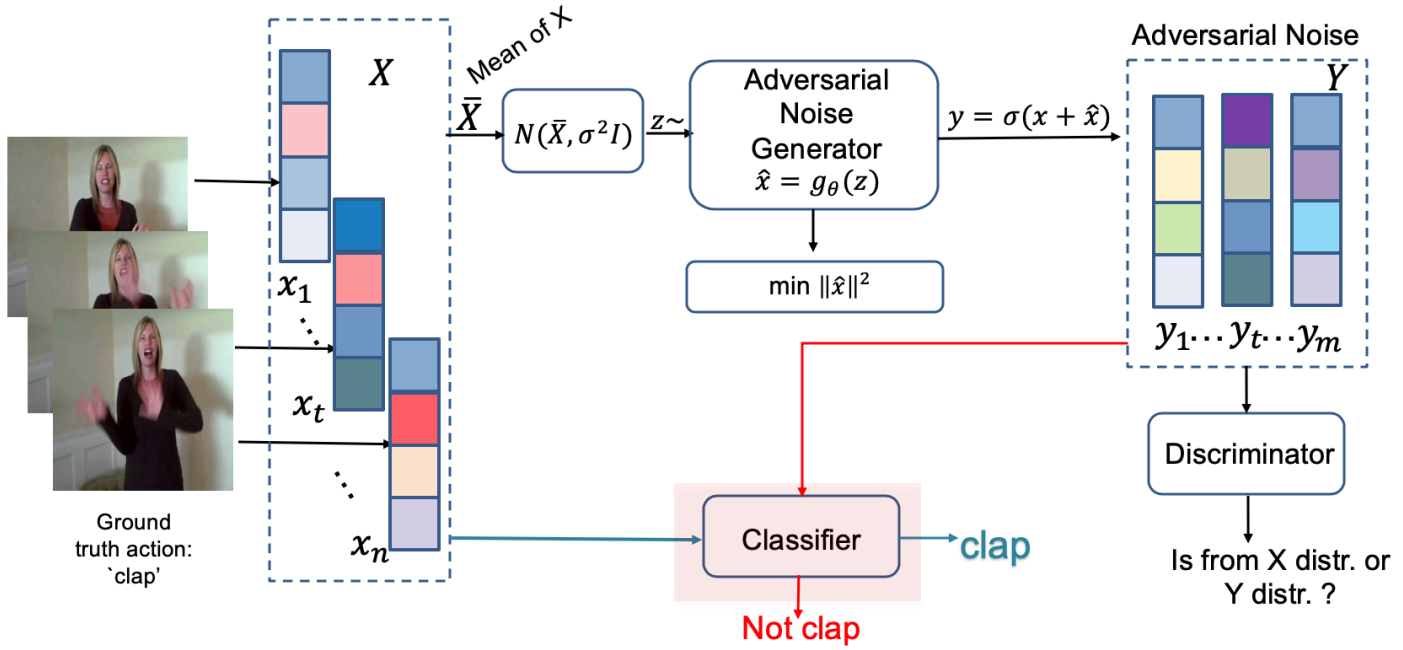
Adversarial Distribution Learning Using Wasserstein GAN



$$\mathcal{L}_A(\theta) = \min_{\theta} \max_{h \in L_1} \mathbb{E}_{\mathbf{x} \in \mathbf{X} \sim \mathcal{D}} [h(\mathbf{x})] - \mathbb{E}_{\substack{\mathbf{y} = \sigma(\mathbf{x} + \hat{\mathbf{x}}), \\ \hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \gamma)}} [h(\mathbf{y})]$$

$$+ \lambda_1 \left(\zeta(\mathbf{x}, \ell_{\mathbf{x}}) - \zeta(\sigma(\mathbf{x} + \hat{\mathbf{x}}), \ell_{\mathbf{x}}) \right) + \lambda_2 \left\| \hat{\mathbf{x}} \right\|^2 .$$

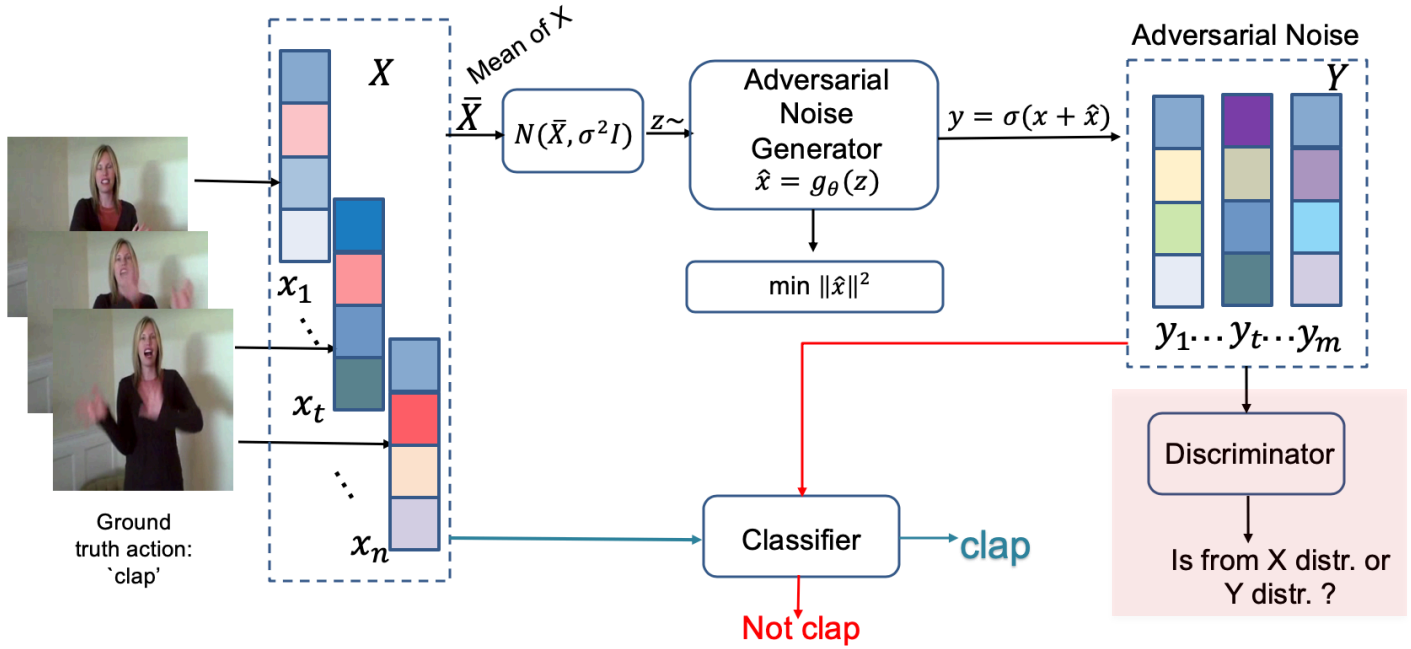
Adversarial Distribution Learning Using Wasserstein GAN



$$\mathcal{L}_A(\theta) = \min_{\theta} \max_{h \in L_1} \mathbb{E}_{\mathbf{x} \in \mathbf{X} \sim \mathcal{D}} [h(\mathbf{x})] - \mathbb{E}_{\substack{\mathbf{y} = \sigma(\mathbf{x} + \hat{\mathbf{x}}), \\ \hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \gamma)}} [h(\mathbf{y})]$$

$$+ \lambda_1 \left(\zeta(\mathbf{x}, \ell_{\mathbf{x}}) - \zeta(\sigma(\mathbf{x} + \hat{\mathbf{x}}), \ell_{\mathbf{x}}) \right) + \lambda_2 \|\hat{\mathbf{x}}\|^2 .$$

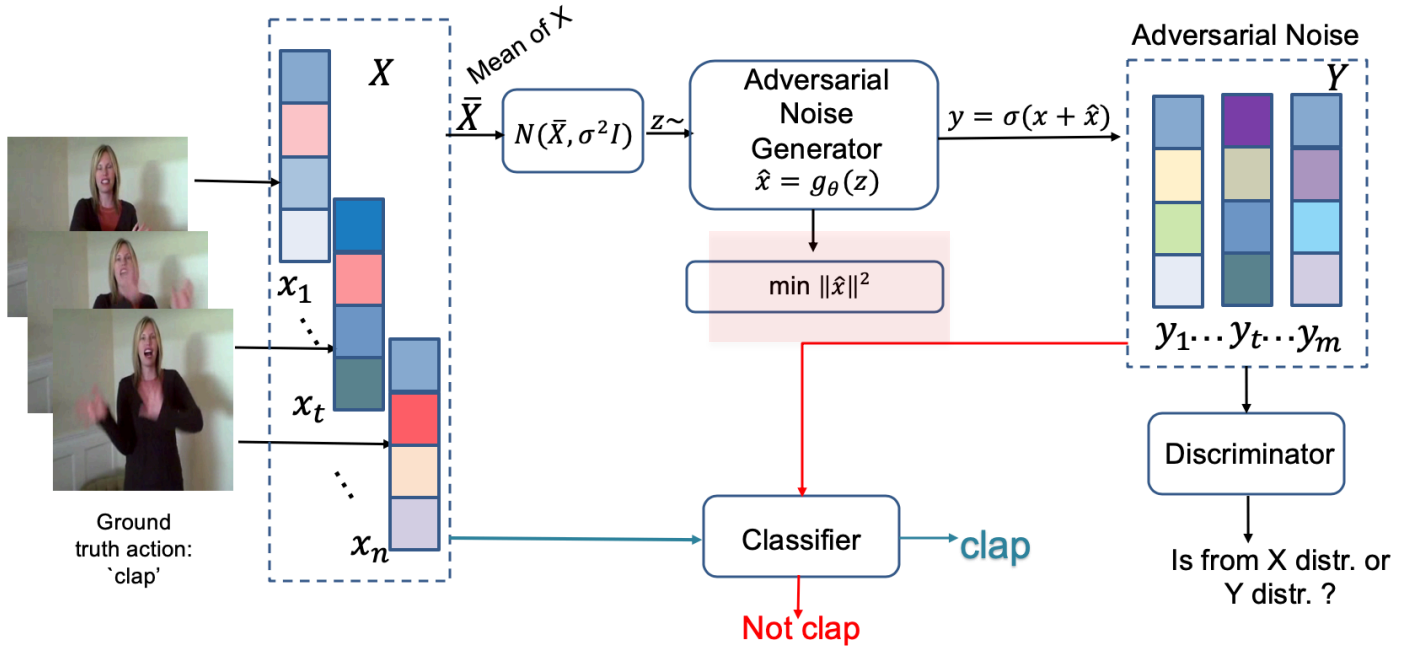
Adversarial Distribution Learning Using Wasserstein GAN



$$\mathcal{L}_A(\theta) = \min_{\theta} \max_{h \in L_1} \mathbb{E}_{\mathbf{x} \in \mathbf{X} \sim \mathcal{D}} [h(\mathbf{x})] - \mathbb{E}_{\substack{\mathbf{y} = \sigma(\mathbf{x} + \hat{\mathbf{x}}), \\ \hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \gamma)}} [h(\mathbf{y})]$$

$$+ \lambda_1 \left(\zeta(\mathbf{x}, \ell_{\mathbf{x}}) - \zeta(\sigma(\mathbf{x} + \hat{\mathbf{x}}), \ell_{\mathbf{x}}) \right) + \lambda_2 \left\| \hat{\mathbf{x}} \right\|^2 .$$

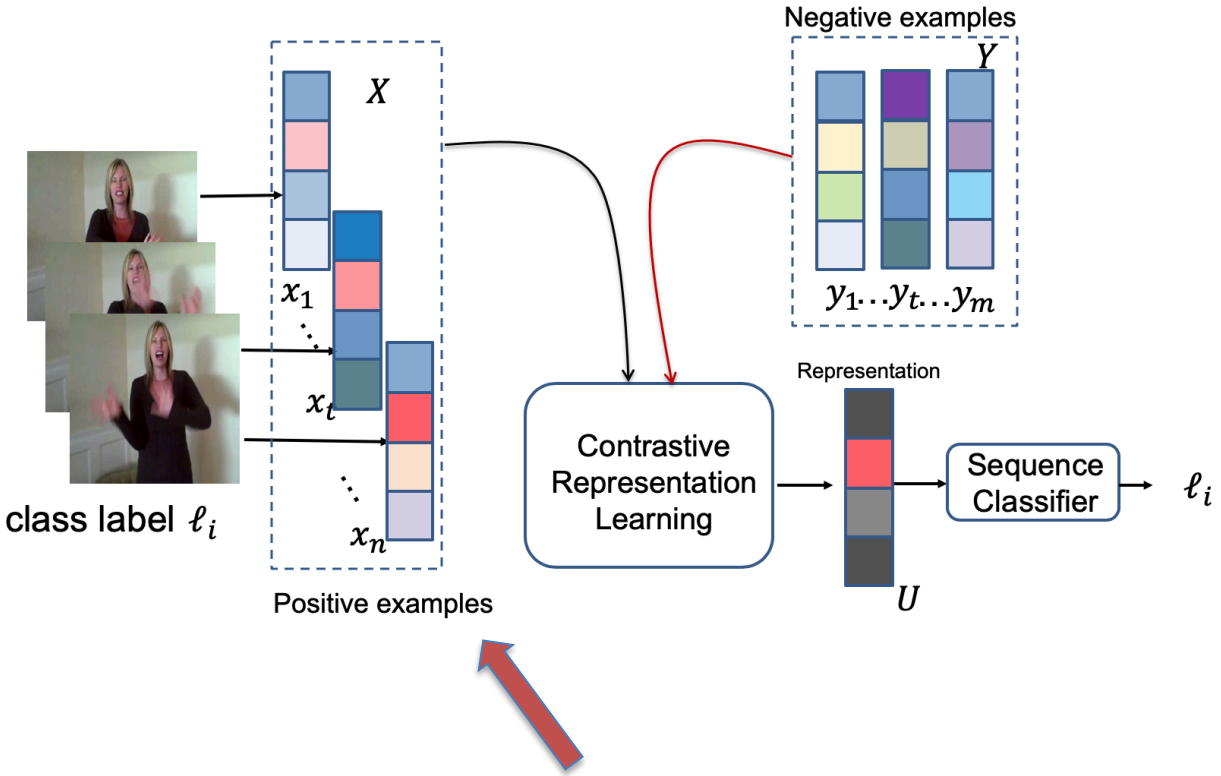
Adversarial Distribution Learning Using Wasserstein GAN



$$\mathcal{L}_A(\theta) = \min_{\theta} \max_{h \in L_1} \mathbb{E}_{\mathbf{x} \in \mathbf{X} \sim \mathcal{D}} [h(\mathbf{x})] - \mathbb{E}_{\substack{\mathbf{y} = \sigma(\mathbf{x} + \hat{\mathbf{x}}), \\ \hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \gamma)}} [h(\mathbf{y})]$$

$$+ \lambda_1 \left(\zeta(\mathbf{x}, \ell_{\mathbf{x}}) - \zeta(\sigma(\mathbf{x} + \hat{\mathbf{x}}), \ell_{\mathbf{x}}) \right) + \lambda_2 \left\| \hat{\mathbf{x}} \right\|^2 .$$

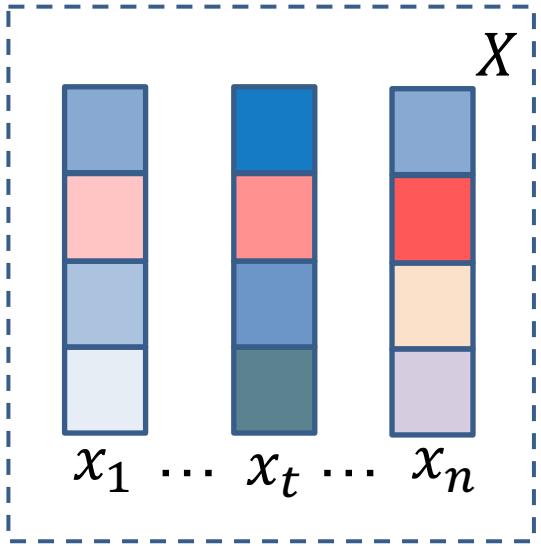
Contrastive Representation Learning



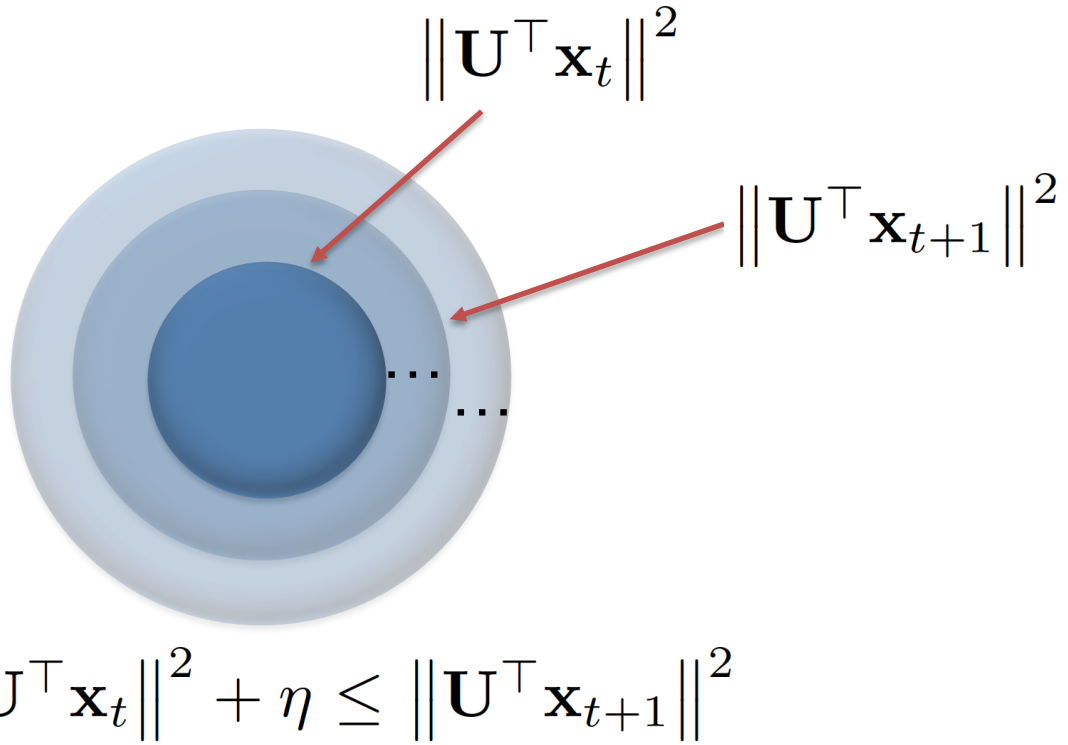
Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?
4. How should we "represent" the sequence?

Capturing Temporal Order

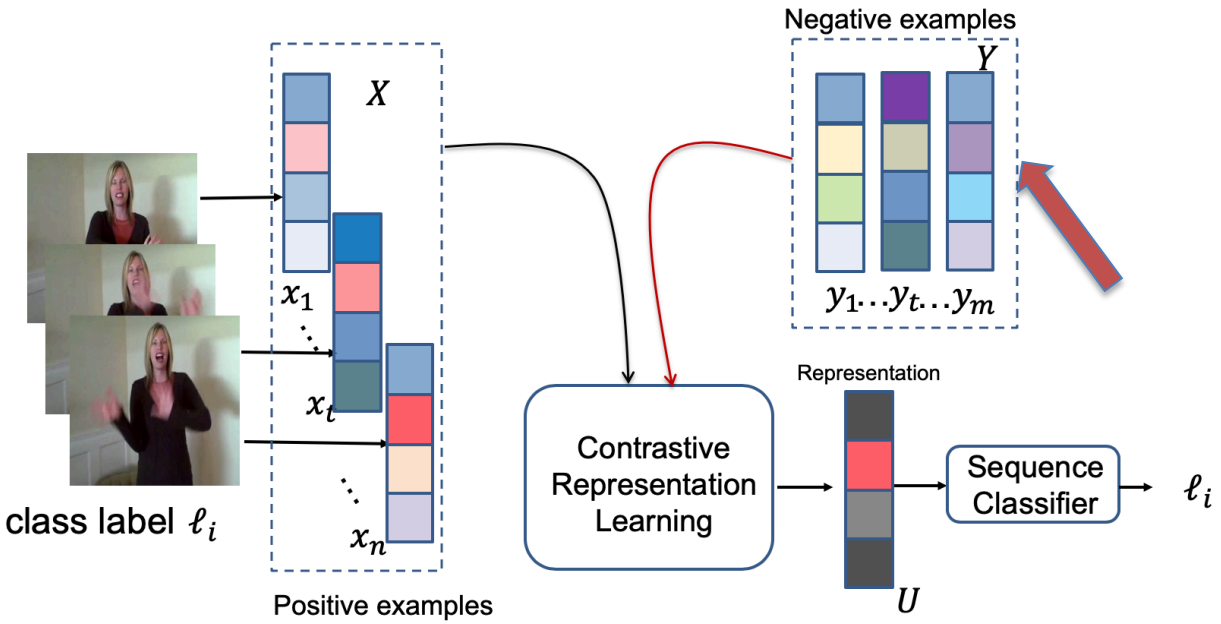


Find U such that



We use Generalized Rank Pooling (Cherian et al., CVPR 2017)
The idea is to find subspaces U
such that projections x on to U is temporally ordered.

Contrastive Representation Learning

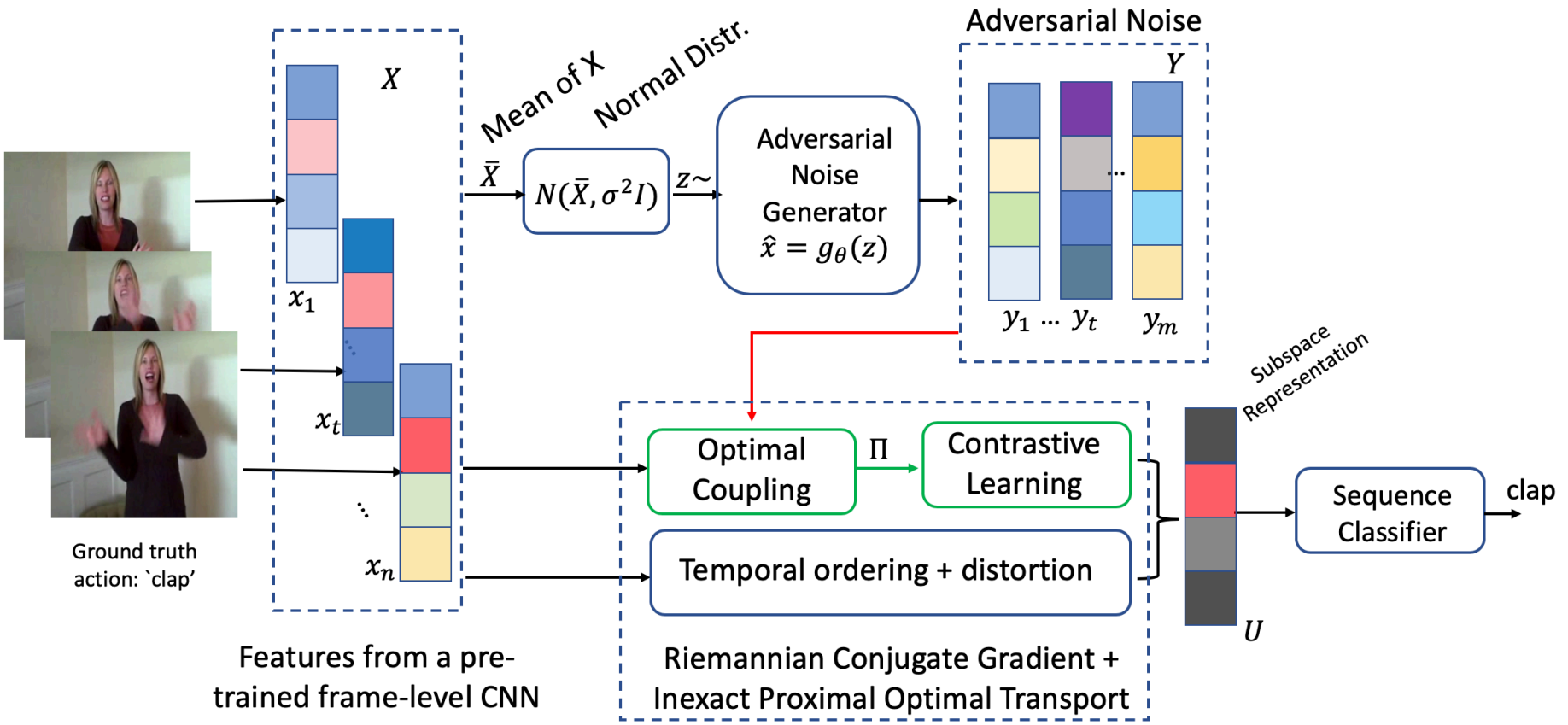


Questions:

1. How should we characterize contrastiveness?
2. How can we find negative examples?
3. How can we capture the temporal order in X?
4. How should we “represent” the sequence?

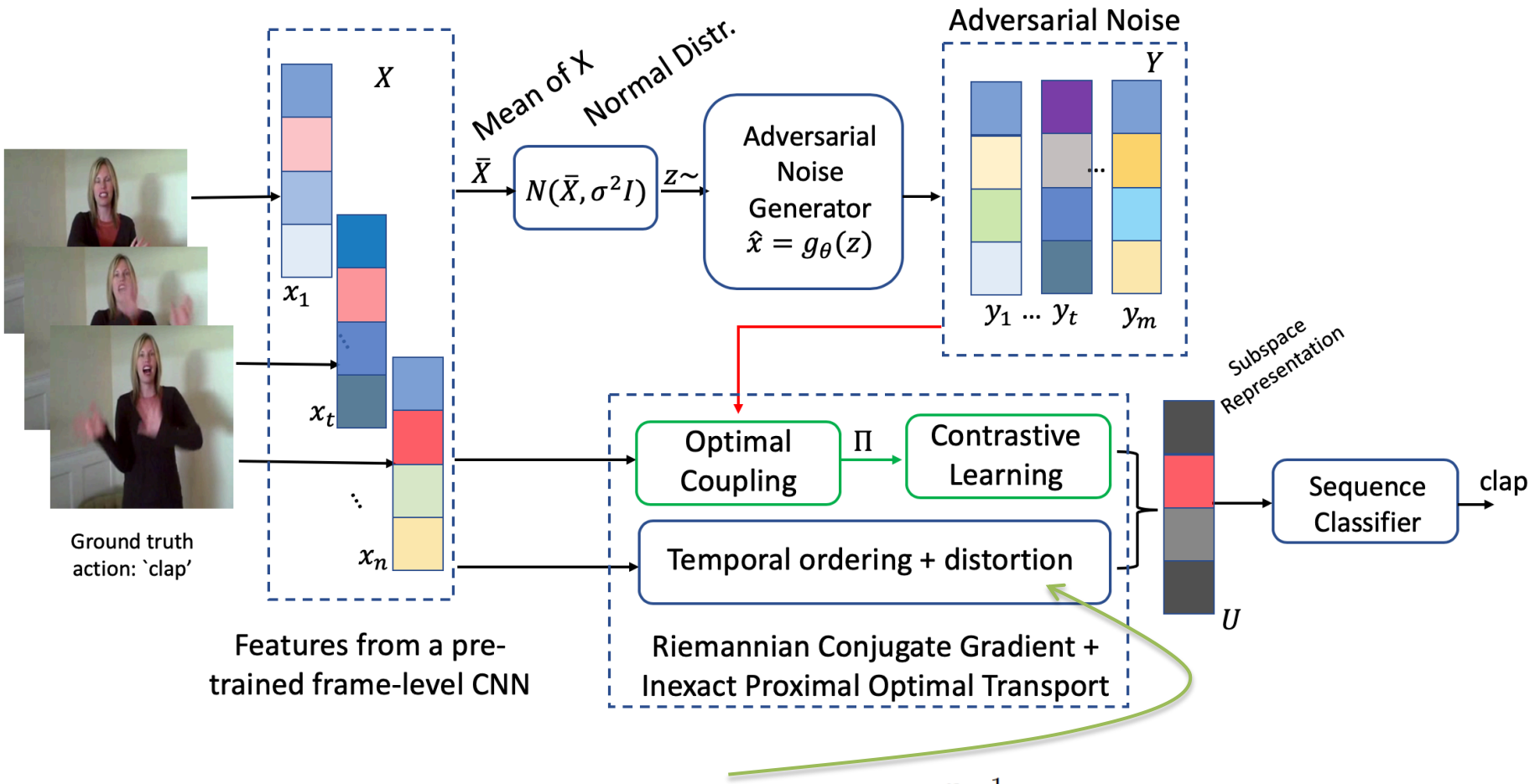
We use the subspace U as the representation of the sequence and use a subspace kernel learning SVM as the sequence classifier.

Putting it all Together!



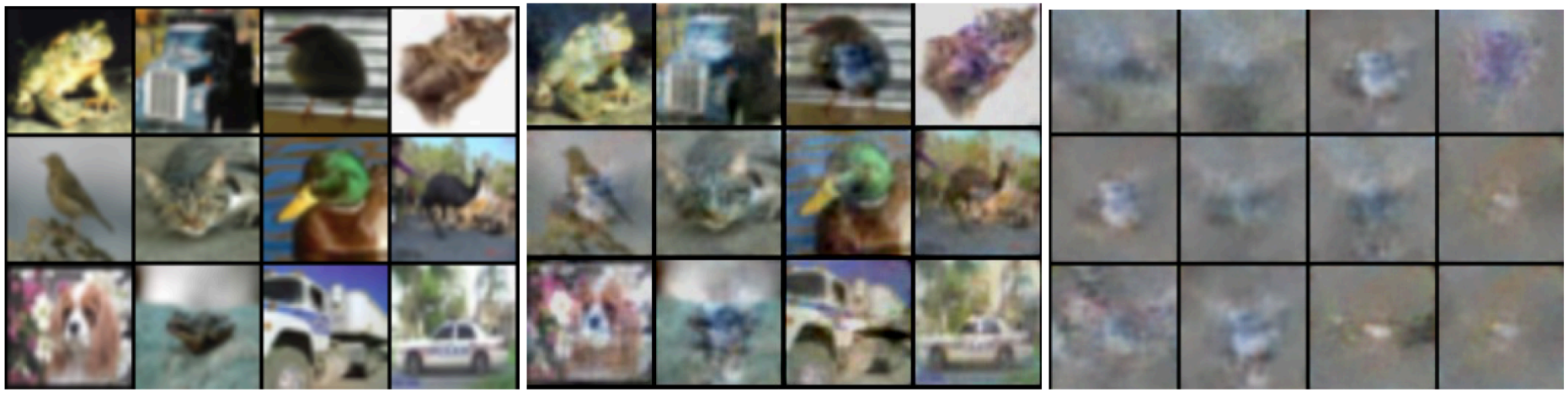
$$\max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathcal{L}_R(\mathbf{U}) := \mathcal{L}_{OT}(\mathbf{U}) - \beta_1 \sum_{\mathbf{x}\in\mathbf{X}} \|f_{\mathbf{U}}(\mathbf{x}) - \mathbf{x}\|^2 - \beta_2 \sum_{t=1}^{n-1} \left[\|\mathbf{U}^\top \mathbf{x}_t\|^2 + \eta - \|\mathbf{U}^\top \mathbf{x}_{t+1}\|^2\right]_+$$

Putting it all Together!



$$\max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathcal{L}_R(\mathbf{U}) := \mathcal{L}_{OT}(\mathbf{U}) - \beta_1 \sum_{\mathbf{x}\in\mathbf{X}} \|f_{\mathbf{U}}(\mathbf{x}) - \mathbf{x}\|^2 - \beta_2 \sum_{t=1}^{n-1} \left[\|\mathbf{U}^\top \mathbf{x}_t\|^2 + \eta - \|\mathbf{U}^\top \mathbf{x}_{t+1}\|^2\right]_+$$

Experiments : Adversarial Noise Generator



CIFAR10 images Corrupted Images Generated noise

The generated noise appears to be targeting regions in the image that are relevant for recognition. Thus, if our representation maximizes the distance between the corrupted and uncorrupted images, it must focus on regions useful for recognition.

Experiments and Results

- We used two action recognition datasets:
  - **JHMDB dataset**
    - 21 classes, ~10-40 frames per sequence, ~900 sequences
    - We used VGG features per frame and I3D features for short clips
  - **HMDB dataset**
    - 51 classes, ~20-400 frames per sequence, ~6000 sequences
    - We used I3D features for short clips

  The I3D network was pre-trained on Kinetics. We did not fine-tune to the above datasets.

Experiments and Results

Ablation	JHMDB (vgg)			JHMDB (I3D)			HMDB (I3D)		
	RGB	FLOW	R+F	RGB	FLOW	R+F	RGB	FLOW	R+F
Avg. Pool	47.0	63.0	73.1	77.5	81.0	85.0	68.2	69.5	76.5
COT + Random	48.0	63.9	77.9	62.2	77.2	79.4	68.5	71.1	72.5
ACOT	49.3	65.0	75.0	76.1	81.2	90.0	69.5	74.6	76.4
ACOT + PCA	49.5	65.7	75.6	77.6	82.8	90.6	69.8	74.9	76.6
AC + PCA + order (No OT)	49.0	66.1	75.8	75.2	80.0	89.8	70.2	74.8	76.3
ACOT + PCA + order	**50.3**	**69.2**	**79.8**	**78.1**	**82.9**	**91.5**	**70.8**	**75.5**	**79.1**

A = Adversarial, C = Contrastive, OT = Optimal Transport, Random = using random noise (instead of adversarial), PCA = Regularization penalty, Order = Temporal ordering

Experiments and Results

Ablation	JHMDB (vgg)			JHMDB (I3D)			HMDB (I3D)		
	RGB	FLOW	R+F	RGB	FLOW	R+F	RGB	FLOW	R+F
Avg. Pool	47.0	63.0	73.1	77.5	81.0	85.0	68.2	69.5	76.5
COT + Random	48.0	63.9	77.9	62.2	77.2	79.4	68.5	71.1	72.5
ACOT	49.3	65.0	75.0	76.1	81.2	90.0	69.5	74.6	76.4
ACOT + PCA	49.5	65.7	75.6	77.6	82.8	90.6	69.8	74.9	76.6
AC + PCA + order (No OT)	49.0	66.1	75.8	75.2	80.0	89.8	70.2	74.8	76.3
ACOT + PCA + order	**50.3**	**69.2**	**79.8**	**78.1**	**82.9**	**91.5**	**70.8**	**75.5**	**79.1**

A = Adversarial, C = Contrastive, OT = Optimal Transport, Random = using random noise (instead of adversarial), PCA = Regularization penalty, Order = Temporal ordering

Experiments and Results

Ablation	JHMDB (vgg)			JHMDB (I3D)			HMDB (I3D)		
	RGB	FLOW	R+F	RGB	FLOW	R+F	RGB	FLOW	R+F
Avg. Pool	47.0	63.0	73.1	77.5	81.0	85.0	68.2	69.5	76.5
COT + Random	48.0	63.9	77.9	62.2	77.2	79.4	68.5	71.1	72.5
ACOT	49.3	65.0	75.0	76.1	81.2	90.0	69.5	74.6	76.4
ACOT + PCA	49.5	65.7	75.6	77.6	82.8	90.6	69.8	74.9	76.6
AC + PCA + order (No OT)	49.0	66.1	75.8	75.2	80.0	89.8	70.2	74.8	76.3
ACOT + PCA + order	**50.3**	**69.2**	**79.8**	**78.1**	**82.9**	**91.5**	**70.8**	**75.5**	**79.1**

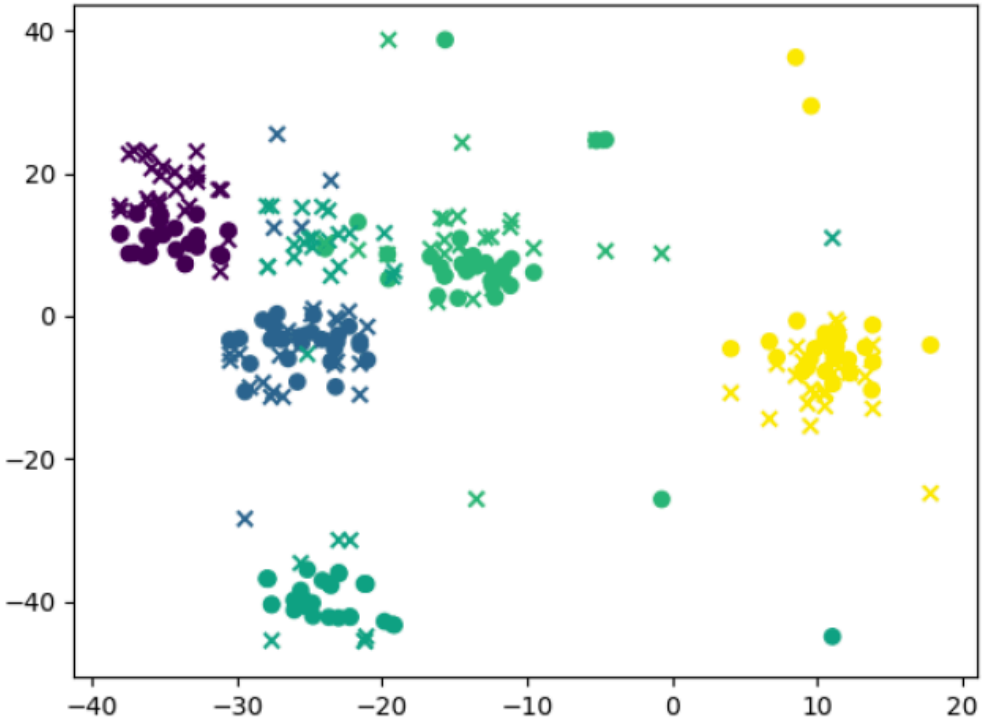
A = Adversarial, C = Contrastive, OT = Optimal Transport, Random = using random noise (instead of adversarial), PCA = Regularization penalty, Order = Temporal ordering

Experiments and Results

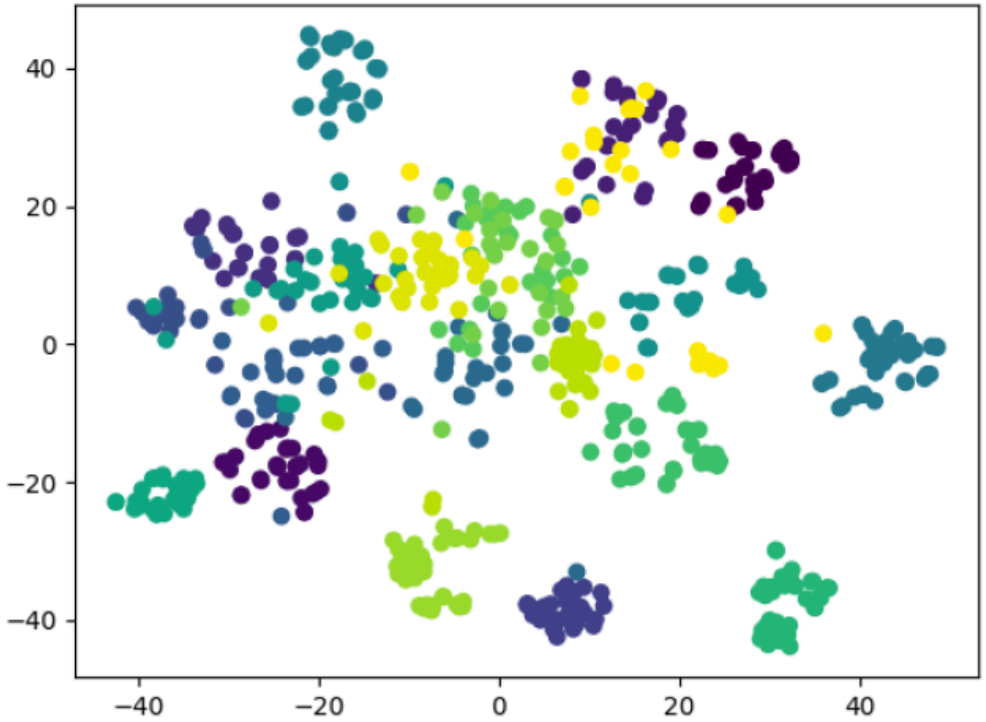
Ablation	JHMDB (vgg)			JHMDB (I3D)			HMDB (I3D)		
	RGB	FLOW	R+F	RGB	FLOW	R+F	RGB	FLOW	R+F
Avg. Pool	47.0	63.0	73.1	77.5	81.0	85.0	68.2	69.5	76.5
COT + Random	48.0	63.9	77.9	62.2	77.2	79.4	68.5	71.1	72.5
ACOT	49.3	65.0	75.0	76.1	81.2	90.0	69.5	74.6	76.4
ACOT + PCA	49.5	65.7	75.6	77.6	82.8	90.6	69.8	74.9	76.6
AC + PCA + order (No OT)	49.0	66.1	75.8	75.2	80.0	89.8	70.2	74.8	76.3
ACOT + PCA + order	**50.3**	**69.2**	**79.8**	**78.1**	**82.9**	**91.5**	**70.8**	**75.5**	**79.1**

A = Adversarial, C = Contrastive, OT = Optimal Transport, Random = using random noise (instead of adversarial), PCA = Regularization penalty, Order = Temporal ordering

Experiments: Representation Quality

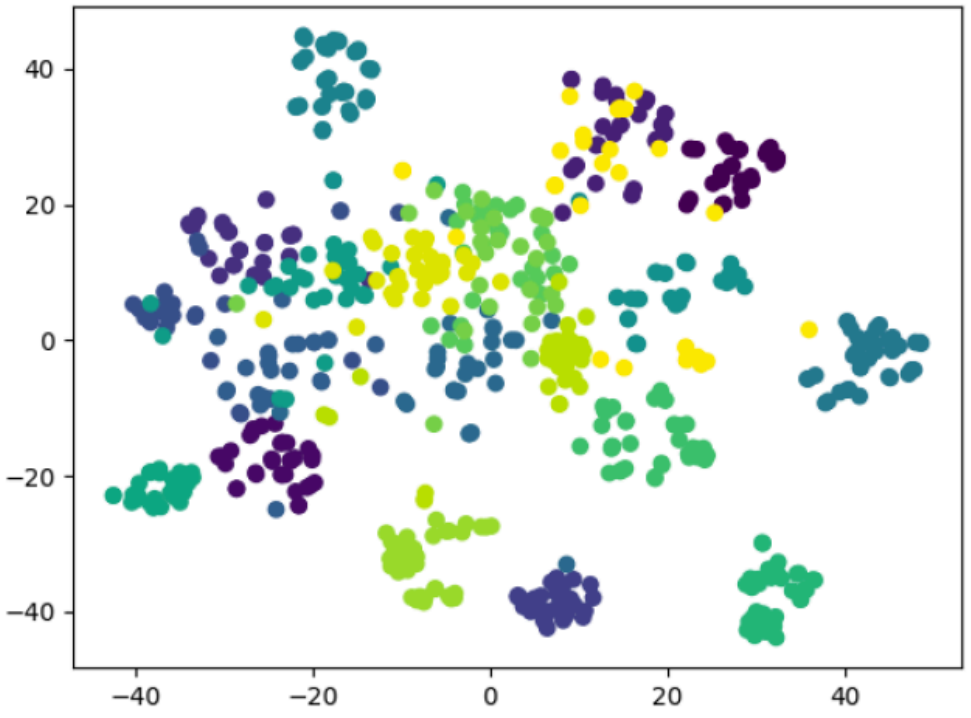


(a) Adv. Samples

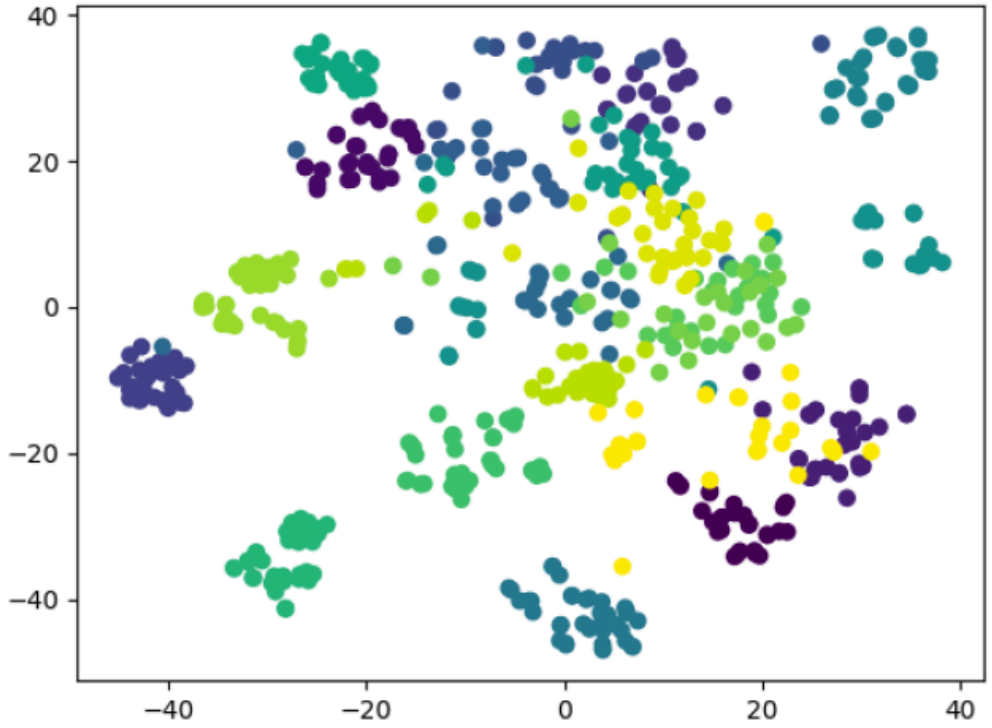


(b) Average Pooling

Experiments: Representation Quality

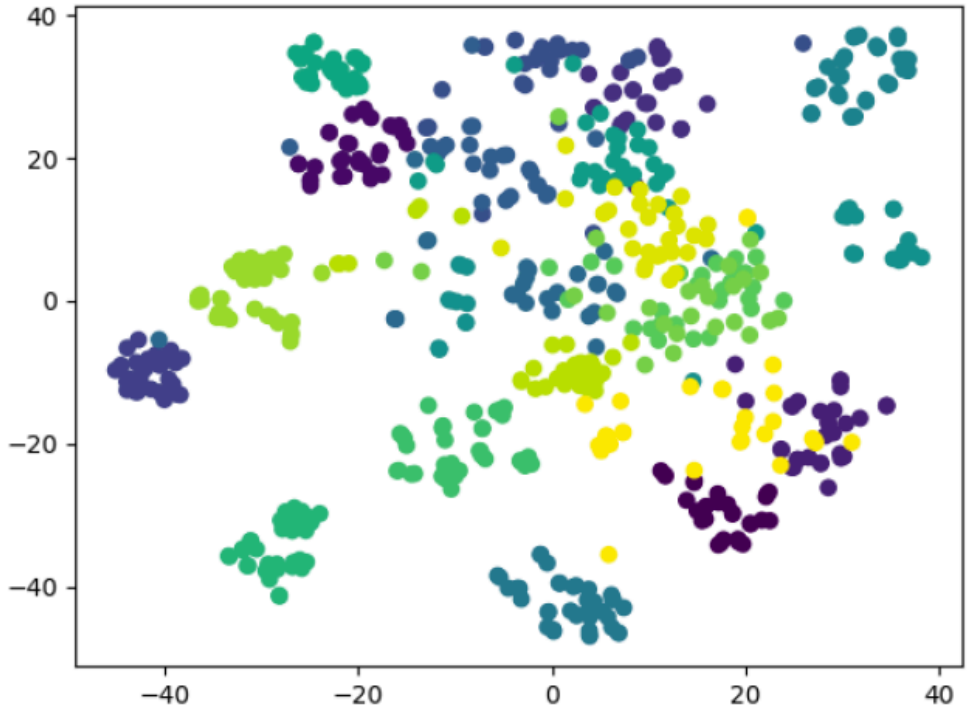


(b) Average Pooling

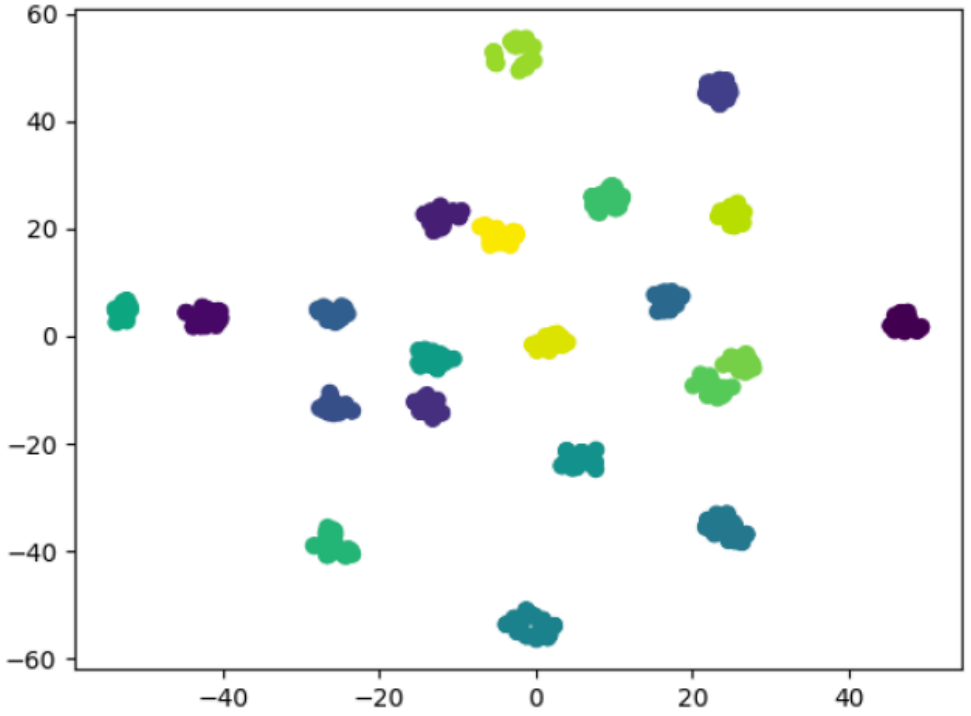


(c) Random Contrastive

Experiments: Representation Quality



(c) Random Contrastive



(d) Adv. Contrastive (Ours)

Experiments and Results

JHMDB using vgg	Accuracy
GRP (Cherian et al., 2017)	70.6
P-CNN (Chéron et al., 2015)	72.2
Kernelized Pooling (Cherian et al., 2018)	73.8
Ours (full model)	**75.7**
JHMDB using 3D-CNNs	Accuracy
Chained (Zolfaghari et al., 2017)	76.1
I3D + Potion (Choutas et al., 2018)	85.5
I3D + Ours (full model)	**87.5**

JHMDB Comparisons

Method	Acc. (%)
I3D (Carreira & Zisserman, 2017)	80.9
Disc. Pool (Wang & Cherian, 2019)	81.3
DSP (Wang & Cherian, 2018)	81.5
Ours (I3D+full model)	**81.8**

HMDB Comparisons

Running time: Excluding time to extract CNN features, our scheme runs at about 30 frames per second in producing the representations.

Conclusions

- We proposed a novel framework for video representation learning by combining
  - Contrastive learning
  - Adversarial distribution learning via GANs, and
  - Optimal transport.

- We used a Riemannian optimization framework for learning our subspace representation.

- Our experiments demonstrate that using adversarially-learned negative examples provide better contrastive learning using the proposed framework.

Our code will be available soon at https://www.merl.com/research/license/