MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Representation Learning via Adversarially-Contrastive Optimal Transport

Cherian, Anoop; Aeron, Shuchin

TR2020-093 July 03, 2020

Abstract

In this paper, we study the problem of learning compact (low-dimensional) representations for sequential data that captures its implicit spatiotemporal cues. To maximize extraction of such informative cues from the data, we set the problem within the context of contrastive representation learning and to that end propose a novel objective via optimal transport. Specifically, our formulation seeks a low-dimensional subspace representation of the data that jointly (i) maximizes the distance of the data (embedded in this subspace) from an adversarial data distribution under the optimal transport, a.k.a. the Wasserstein distance, (ii) captures the temporal order, and (iii) minimizes the data distortion. To generate the adversarial distribution, we propose a novel framework connecting Wasserstein GANs with a classifier, allowing a principled mechanism for producing good negative distributions for contrastive learning, which is currently a challenging problem. Our full objective is cast as a subspace learning problem on the Grassmann manifold and solved via Riemannian optimization. To empirically study our formulation, we provide experiments on the task of human action recognition in video sequences. Our results demonstrate competitive performance against challenging baselines.

International Conference on Machine Learning (ICML)

Representation Learning via Adversarially-Contrastive Optimal Transport

Anoop Cherian [1] **Shuchin Aeron** [2]

Abstract

In this paper, we study the problem of learning compact (low-dimensional) representations for sequential data that captures its implicit spatio-temporal cues. To maximize extraction of such informative cues from the data, we set the problem within the context of contrastive representation learning and to that end propose a novel objective via optimal transport. Specifically, our formulation seeks a low-dimensional subspace representation of the data that jointly (i) maximizes the distance of the data (embedded in this subspace) from an adversarial data distribution under the optimal transport, a.k.a. the Wasserstein distance, (ii) captures the temporal order, and (iii) minimizes the data distortion. To generate the adversarial distribution, we propose a novel framework connecting Wasserstein GANs with a classifier, allowing a principled mechanism for producing good negative distributions for contrastive learning, which is currently a challenging problem. Our full objective is cast as a subspace learning problem on the Grassmann manifold and solved via Riemannian optimization. To empirically study our formulation, we provide experiments on the task of human action recognition in video sequences. Our results demonstrate competitive performance against challenging baselines.

1. Introduction

Recent advancements in deep neural network architectures (Greff et al., 2016; Merity et al., 2018; Sutskever et al., 2014; Zilly et al., 2017; Varol et al., 2017; Chung et al., 2015; Kim et al., 2019) have resulted in significant progress towards our ability to model and reason over sequential data. However, this problem is far from considered solved and continues to be challenging, especially in high-dimensional spatio-temporal settings. There are several practical issues that lead to this difficulty, notably (i) most of the highly successful neural network models operate on data of a fixed length (such as images), however temporally-evolving data, such as for example the frame-level features from a video recognition task, could be of arbitrary temporal length, and (ii) the data may be entangled with nuisance factors, such as for example, features corresponding to temporally-evolving background clutter, which may make inference difficult. While, it may be possible to extend popular deep architectures to address these challenges, they may be computationally heavy or require large-scale annotated data, which may be difficult to gather. We refer the reader to several papers (Dollar et al., 2005; Varol et al., 2016; Wang & Gupta, 2015; Tran et al., 2017; Wu et al., 2017; Girdhar et al., 2019) illustrating the wide range of approaches undertaken to tackle this challenging problem.

In this paper, we address these issues by learning a compact representation of a given video sequence of arbitrary length, that maximally captures the spatio-temporal information, while at the same time can be effectively fed to a light weight classifier for action recognition. We approach this representation learning problem from the perspective of contrastive learning (Saunshi et al., 2019; van den Oord et al., 2018; Bose et al., 2018; Tschannen et al., 2019; Wang & Cherian, 2018) that has recently emerged as a flexible yet powerful tool for learning generic representations for high dimensional data, using positive and negative examples. The key idea in contrastive learning is to produce a representation that is closer to the positive (given data) examples, and farther from the negatives. Usually, the negative examples are randomly picked. However, it is usually seen that the performance of these algorithms heavily depend on the choice of the negatives and their *pairings* with the positives. (Arora et al., 2019; Bose et al., 2018).

Suppose we are given a set of negative examples to work with. As the performance of contrastive learning depends on the coupling between positive and negative examples, to this end, we propose a novel contrastive learning objective that *simultaneously optimizes* for learning a representation via a tightly coupled interplay between four key components, namely (i) generating a (probabilistic/soft) coupling with the negative data via *minimizing* the optimal transport /Wasserstein distance (Santambrogio, 2015) between the

[1] Mitsubishi Electric Research Labs, Cambridge, MA. [2] Tufts University, Medford, MA. Correspondence to: Anoop Cherian <cherian@merl.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

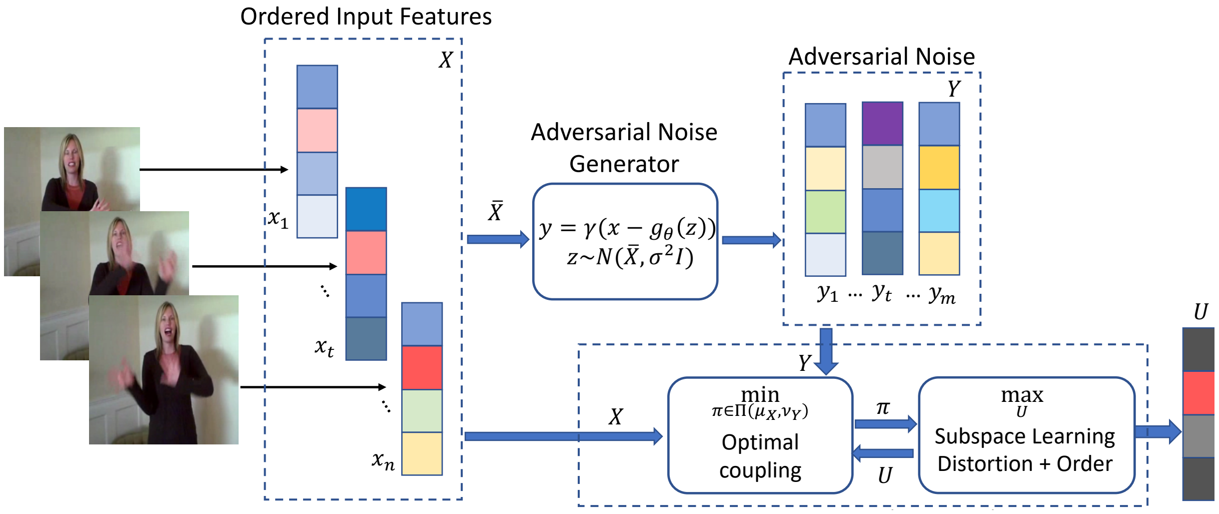

Ordered Input Features
X
x_1
x_t
x_n
$\bar{X}$
Adversarial Noise Generator
$y = \gamma(x - g_\theta(z))$
$z \sim N(\bar{X}, \sigma^2 I)$
Adversarial Noise
Y
$y_1 \cdots y_t \cdots y_m$
Y
X
$\min_{\pi \in \Pi(\mu_X, \nu_Y)}$
Optimal coupling
π
U
$\max_U$
Subspace Learning
Distortion + Order
U

Figure 1. Our overall architecture. The input frames are first encoded into a set $\mathbf{X}$ of ordered feature vectors $\mathbf{x}_t$ using a (pre-trained) neural network. These features are then used in an adversarial noise generator (implemented using a Wasserstein GAN trained for adversarial losses) to generate a set $\mathbf{Y}$ of adversarial noise samples. Next, we use $\mathbf{X}$ and $\mathbf{Y}$ in a joint optimal transport and representation learning formulation that tries to (i) minimize the optimal coupling π between the two sets, while (ii) also learn a subspace $\mathbf{U}$ that maximizes the distance between projections of $\mathbf{X}$ onto $\mathbf{U}$ and the adversarial noise $\mathbf{Y}$. This latter cost also includes distortion and ordering penalties. Illustratively, we depict the useful dimensions of the input in 'red' color (or its variants). The idea is that the representation $\mathbf{U}$ can filter this dimension via contrasting against the noise.

positive and negative examples, (ii) a representation learning cost that seeks a low-dimensional data subspace such that the projections of the data onto this subspace *maximizes* their distance from the coupled negative examples, (iii) a *distortion* penalty that prevents the subspace projections from diverging too much from the input data, and (iv) an ordering constraint on the subspace projections that captures the sequentiality of the input.

Having outlined our novel approach for simultaneous learning of coupling between the positive and negative examples and the (contrastive) representation, we turn towards addressing generation of good negative examples. As pointed out in (Tschannen et al., 2019), design of good negative distribution and its impact on contrastive learning is not fully addressed yet. In contrast to existing works such as, (Hénaff et al., 2019; Oord et al., 2018; Sun et al., 2019; Wang & Gupta, 2015) we show that, when coupled with the objective to learn a contrastive representation, one must allow a high discrimination, with respect to the intended task, from the positive distribution while maintaining good correlation.

We resolve these conflicting objectives in learning the contrastive distribution by trading off conditional distributional discrepancy with a (given) classifier accuracy on the contrastive distribution. Specifically, we use a Wasserstein GAN that takes as input positive examples from the dataset, and learn to generate new samples (conditioned on the positives), where these generated samples when added to their respective positive examples will ensure two properties, namely (i) the modified positives must be as close in distribution as possible to the true positives in the dataset, and (ii) a classifier that has good performance on the true positives must have high misclassification rate on these modified positives. Figure 1 pictorially depicts our complete framework.

We summarize our novel contributions below.

1. We propose a representation learning cost that *naturally* blends contrastive learning, adversarial learning, optimal transport, and Riemannian geometry into one framework.

2. We show that our objective for learning contrastive representation, while completely differing in its aims, is related to the subspace robust optimal transport distances proposed in (Paty & Cuturi, 2019). We characterize this relation in Theorem 1, thereby making a novel connection between contrastive learning and robust optimal transport.

3. Further, we present an adversarial distribution learning setup within which, it is possible to optimize over and regulate the choice of distribution for negative samples that is known to be critical for contrastive representation learning.

4. We apply this framework for the problem of learning representations for classifying action video sequences and obtain promising classification results.

2. Related Work

Our approach has several parallels and similarities with existing works that we systematically outline below. Contrastive estimation (Smith & Eisner, 2005) and noise-contrastive estimation (Gutmann & Hyvärinen, 2010) are popular representation learning methods that learn via an objective that contrasts data likelihood under the model against the likelihood of the noise or implicitly-constructed contrastive (i.e., negative) examples. Popular deep metric learning and triplet losses (Hoffer & Ailon, 2015) can also be considered as variants of contrastive learning when explicit pairwise relationships between data samples are available. As empirically shown in many works and recently theoretically argued in (Arora et al., 2019), contrastive learning can reduce the sample complexity for downstream tasks.

There have been recent efforts at unifying the general representation learning methods and contrastive methods (Hoffer & Ailon, 2015; Arora et al., 2019), such as for example via the maximum InfoNCE principle (Tschannen et al., 2019; Oord et al., 2018), in which the main idea is to implicitly maximize the mutual information (Cover & Thomas, 2006) between the learned representation and the original data. However, these formulations assume a good approximation of mutual information from the given samples – a problem that is typically hard. Recently, using different forms of variational characterizations of mutual information (Poole et al., 2019), efficient estimators have been proposed. Nevertheless, the resulting optimization and training may become unstable (Song & Ermon, 2019).

Among works based on maximizing InfoNCE, one similar to ours is (Oord et al., 2018) that proposes contrastive predictive coding through a density ratio capturing the mutual information between the representation and future samples of the sequence. They use noise contrastive estimation on their representation learning cost, where the noise distribution is considered as those plausibly unrelated to the input. Our proposed framework is different from theirs in that we explicitly seek a negative distribution that can potentially increase the contrastiveness of useful cues via learning to generate these hard negative samples. Attempts towards improving the negative sampler in contrastive learning have been made in (Bose et al., 2018). Here the authors propose to use a mixture of unconditional and conditional negative distributions, conditioned on given data, and parametrized via an implicit generative model; however with a totally different loss function than ours. In contrast, we achieve the required discrimination using a classifier, while our proposed variant of WGAN captures the similarity of the negatives to the given data.

Our work is also related to discriminative pooling (Wang & Cherian, 2019) that proposes to generate negative samples via passing random noise through an image-trained CNN, however is not adversarial. In (Wang & Cherian, 2018), the authors propose an adversarial setup in a discriminative representation learning framework, however uses a deterministic deep model to learn a single adversarial sample per data point. Instead, our proposed framework can generate distributions of adversarial samples. Another paper related to ours is (Cherian et al., 2017) that proposes to use subspace representations for video sequences. While, we also use their temporal learning constraints, our optimal transport and adversarial distribution learning offer a richer representation learning setup via suppressing perhaps false-positive temporal features, such as temporally-evolving noise.

3. Problem Formulation

In this section, we describe our problem setup, introduce our notation, and review some prior work on which we build our proposed algorithms. Following standard notation, we use uppercase boldface letters $\mathbf{X}$ to denote matrices, and lowercase boldface $\mathbf{x}$ to denote vectors. Refer Figure 1 for contextualizing our notation and variables in the sequel.

Suppose we are given a set of N data sequences $\mathcal{D} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$, where each $\mathbf{X}_i = \left\langle \mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n_i}^i \right\rangle$ is a sequence of n_i ordered feature vectors and each $\mathbf{x}_t \in \mathbb{R}^d$. Further, we assume $\mathbf{X}_i$ is associated with a ground truth class label $\ell_i \in \mathcal{L}$; $\mathcal{L}$ denoting a given set of labels. We also assume each $\mathbf{x} \in \mathbf{X}$ is an independent sample from a data distribution $P_{\mathcal{D}}(\overline{\mathbf{X}})$ conditioned on the mean $\overline{\mathbf{X}}$ of the sequence $\mathbf{X}$.[1] Note that we do not make any explicit assumption on what these sequences represent. For example, in a video recognition application, each $\mathbf{x}_t$ could be the output of a frame-level deep neural network that is trained on individual frames against their respective video label (Simonyan & Zisserman, 2014).

As each feature $\mathbf{x}_t$ in a sequence $\mathbf{X}$ is assumed to be generated independently without accessing the rest of the sequence, these individual features could be noisy; for example, they could be entangled with irrelevant features from the background. Our key assumption is that the *useful* temporally-evolving features belong to subspaces in this d-dimensional feature space. Thus, our main idea is to design an objective that could extract these subspaces in a compact form, denoted $\mathbf{U}(\mathbf{X})$, such that by using this representation some suitable empirical sequence recognition loss $\mathcal{L}_{\mathcal{D}}$ is minimized, where:

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

The loss $\mathcal{L}_{\mathcal{D}}$ aggregates the error $\mathcal{L}_C$ in training a classifier C on the representations $\mathbf{U}_i$ for each sequence against its

[1] We may use any other central tendency of the sequence to define this distribution.

ground truth label ℓ_i. Further, the $\mathbf{U}_i$'s are obtained via optimizing a sequence level representation learning objective captured by the loss $\mathcal{L}_R$. In a classical feature learning setup (Simonyan & Zisserman, 2014; Carreira & Zisserman, 2017), $\mathcal{L}_R$ finds a vector $\mathbf{U}$ that minimizes, say, the mean-squared error to the data samples; which boils down to the average feature. Thus, in that case, the $\arg\min$ optimization is merely the average pooling scheme. In this paper, we generalize this pooling for richer and better representation learning. While, we can easily train for the two losses $\mathcal{L}_C$ and $\mathcal{L}_R$ jointly in an end-to-end manner (Wang & Cherian, 2019), in this work, we deal with them separately so that we have better control of each of them. In the next few sections, we look deeper into the representation loss using a contrastive learning framework. We will describe the classifier loss $\mathcal{L}_C$ in Sec. 3.6

3.1. Contrastive Learning via Optimal Transport

Suppose, we treat a sequence $\mathbf{X}_i$ as a *set* of positive examples (ignoring the order within), and assume that we have access to a set of negative examples, denoted $\mathbf{Y}_i = \{\mathbf{y}_1^i, \mathbf{y}_2^i, \cdots, \mathbf{y}_m^i\}$, each $\mathbf{y} \in \mathbb{R}^d \sim P_{\mathbf{Y}}$. In noise contrastive estimation, $P_{\mathbf{Y}}$ is typically assumed to be either uniform or Gaussian noise. The goal of contrastive learning in this setup is to find a suitable representation for $\mathbf{X}$ that is maximally "distant" from $\mathbf{Y}$. How should we characterize this contrastiveness? Given that we are working with sets of data points under the assumption that they are random samples from underlying probability distributions, a natural possibility is to consider the Optimal Transport (OT), also known as the Wasserstein distance between the two distributions (Santambrogio, 2015).

Recall that the Wasserstein distance, denoted by $W_c(\mu, \nu)$, between two probability measures μ, and ν that are both supported on $\mathbb{R}^d$ with respect to a ground cost $c(\mathbf{x}, \mathbf{y}), \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, is given by (Santambrogio, 2015):

$$W_c(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\, c(\mathbf{x}, \mathbf{y}), \tag{2}$$

where $\Pi(\mu, \nu)$ denotes the set of all couplings (joint probability distributions) with marginals μ and ν. Typically, in the absence of any other information and when the measures are discrete, μ, ν are chosen to be uniform distributions over the support. Specific to our case, given positive examples $\mathbf{X}$, we let $\mu_{\mathbf{X}}$ be the empirical distribution (with equal, i.e., uniform probability) over $\mathbf{x}_t \in \mathbf{X}$, i.e., $\mu_{\mathbf{X}} = \sum_{t=1}^n \frac{1}{n}\delta(\mathbf{x}_t)$, $\delta(\mathbf{x}_t)$ denoting the Dirac measure at $\mathbf{x}_t$. Similarly, let $\nu_{\mathbf{Y}}$ be the empirical distribution of the negative samples $\mathbf{Y}$, also uniform over the samples.

Next, let $f_{\mathbf{U}} : \mathbb{R}^d \to \mathbb{R}^d$ be a mapping parameterized by the *to be learned* representation $\mathbf{U}$. Then, we formulate our constrastive optimal transport problem for representation learning as:

$$\max_{\mathbf{U}}\ \mathcal{L}_{OT}(\mathbf{U}) := W_c(f_{\mathbf{U}\#}\mu_{\mathbf{X}}, \nu_{\mathbf{Y}}). \tag{3}$$

The notation $f_{\mathbf{U}\#}\rho$ denotes the push-forward measure of ρ under the mapping $f_{\mathbf{U}}$. In this paper, we assume the useful features belong to linear feature subspaces[2] and thus use the mapping f defined as $f = \mathbf{U}\mathbf{U}^\top$ for orthonormal $\mathbf{U} \in \mathbb{R}^{d\times k}$, i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_k$, where $\mathbf{I}_k$ denotes the $k \times k$ identity matrix and $k \ll d$. Rewriting the Wasserstein distance (2) in empirical form, combining it with the definition of f in (3), and using the ℓ_2-norm for the OT cost c, we can write the contrastive representation learning objective as:

$$\max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathcal{L}_{OT}(\mathbf{U}) := \inf_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \sum_{i,j} \pi_{ij} \|f_{\mathbf{U}}(\mathbf{x}_i) - \mathbf{y}_j\|, \tag{4}$$

In the formulation (4), we assume $\mathbf{U} \in \mathcal{G}(d, k)$, the Grassmann manifold of all k-dimensional subspaces of $\mathbb{R}^d$. Recall that $\mathcal{G}(d, k)$ denotes the quotient space $\mathcal{S}(d, k)/\mathcal{O}(k)$ of all $d \times k$ orthonormal matrices $\mathcal{S}(d, k)$ that are invariant to right rotations. Given that our loss $\mathcal{L}_{OT}(\mathbf{U}) = \mathcal{L}_{OT}(\mathbf{U}R)$ for any $k \times k$ orthogonal matrix R, casting the learning objective on the Grassmann manifold is a natural choice.

A question with regard to (4) is why we have $f_{\mathbf{U}}$ acting only on the positive samples? This is because, we assume that the negative samples (as described in Sec. 3.2) share all the subspaces of the positives, except for those subspaces containing relevant cues for classification (e.g., action-related), which are present only in the positives. Thus, asking for a maximal common projection subspace on positive and negatives may lead the optimizer to move away from finding the useful contrastive subspaces in the positives.

3.1.1. Connections to Subspace Robust OT

We note that (4) is itself novel for contrastive learning in that instead of looking for pairs of positive and negative examples as is common in contrastive learning setup, it implicitly learns the best coupling (via the optimal transport plan) and enforces this coupling to be the maximal. Our formulation (4) is related to the recently proposed subspace robust Wasserstein distances (Paty & Cuturi, 2019), which can be defined in our problem setup as:

$$\mathcal{P}_k^2 \triangleq \max_{\mathbf{U}:\mathcal{S}(d,k)}\ \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}}))} \mathbb{E}_\pi \|\mathbf{U}^\top\mathbf{x} - \mathbf{U}^\top\mathbf{y}\|^2, \text{ and}$$

$$\mathcal{S}_k^2 \triangleq \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})}\ \max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathbb{E}_\pi \|\mathbf{U}^\top\mathbf{x} - \mathbf{U}^\top\mathbf{y}\|^2.$$

Suppose, $\mathcal{C}_k^2$ denotes the Wasserstein-2 distance variant of our objective in (4); i.e.,

$$\mathcal{C}_k^2 = \max_{\mathbf{U}\in\mathcal{G}(d,k)}\ \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \sum_{i,j} \pi_{ij} \|f_{\mathbf{U}}(\mathbf{x}_i) - \mathbf{y}_j\|^2. \tag{5}$$

[2] This *linearity* choice is inspired by the observation that usually these deep features are extracted from the penultimate layers of a CNN, subsequently classified using a linear classifier.

It is clear that our subspaces $\mathbf{U}$ act only on the positive examples, and seek the maximal robustness against the chosen negative distribution. The following theorem characterizes the connection between the solutions to the two objectives. The proof can be found in the supplementary material.

Theorem 1. *Assuming n_Y negative samples,*

$$\mathcal{P}_k^2 \le \mathcal{C}_k^2 \le \mathcal{S}_k^2 + \max_{\mathbf{U}\in\mathcal{G}(d,k)} \frac{1}{n_Y}\sum_{j=1}^{n_Y} \|(\mathbf{I}_d - \mathbf{U}\mathbf{U}^\top)\mathbf{y}_j\|^2,$$

with equality iff $k = d$. The variational term on the RHS is equal to the sum of $d-k$ largest eigenvalues of the Gram matrix $\mathbf{\Sigma_Y} = \frac{1}{n_Y}\sum \mathbf{y}_j\mathbf{y}_j^\top$.

3.2. Generating Noise Distributions

As alluded to above, in contrastive learning, typically the noise distribution is assumed uncorrelated to the data. However, it is often observed that as the noise is closer to the data (hard negatives), the quality of the learned representation improves. In this regard, there are two difficulties to address with regard to the noise generation, (i) how to generate the noise distribution ν_Y closer to the data distribution μ_X, and (ii) how to ensure the generated noise will not impact the useful features in $\mathbf{X}$? We resolve this dilemma via generative adversarial networks with some new regularizations.

Concretely, suppose our measure on the negative samples $\nu_{\mathbf{Y}}$ is defined via an implicit function $g_\theta : \mathbb{R}^d \to \mathbb{R}^d$, where θ defines its parameters to be learned. That is, we assume $\mathbf{y} = g_\theta(\mathbf{z})$, where $\mathbf{z} \sim \mathcal{N}(\overline{\mathbf{X}}, \sigma^2 I)$. Specifically, we assume the input to our implicit generator g_θ comes from *normally* distributed data with mean $\overline{\mathbf{X}}$ and standard deviation σ. Note that $\overline{\mathbf{X}}$ defines the average of the samples in the respective sequence, i.e., $\overline{\mathbf{X}} = \frac{1}{n}\sum_t \mathbf{x}_t$. Recall that we had originally assumed each $\mathbf{x} \in \mathbf{X} \sim P_{\mathcal{D}}(\overline{\mathbf{X}})$ (in Sec 3). Now, our goal is to learn θ such that it emulates $P_{\mathcal{D}}(\overline{\mathbf{X}})$, $\forall \mathbf{X} \in \mathcal{D}$, while also capturing other desirable properties listed above. Substituting the definition of $\mathbf{y}$ in (4), we have a modified OT problem:

$$\mathcal{L}'_{OT} := \min_{\theta} \min_{\pi\in\Pi(\mu_{\mathbf{X}}, g_\theta(\overline{\mathbf{X}},\sigma))} \mathbb{E}_\pi \|f_{\mathbf{U}}(\mathbf{x}) - \mathbf{y}\|, \qquad (6)$$

where we use the succinct notation $g_\theta(\overline{\mathbf{X}}, \sigma)$ to explicitly show the relation of the noise distribution ν_Y with the input sequence $\mathbf{X}$. Using Wasserstein-1 distance, we can use the Kantorovich duality to derive a dual form of this objective via learning a 1-Lipschitz function h, and rewriting (6) as:

$$\mathcal{L}''_{OT} := \min_{\theta} \max_{h\in L_1} \mathbb{E}_{\mathbf{x}\sim\mu_{\mathbf{X}}}\left[h(f_{\mathbf{U}}(\mathbf{x}))\right] - \mathbb{E}_{\mathbf{y}\sim g_\theta(\overline{\mathbf{X}},\sigma)}\left[h(\mathbf{y})\right]. \qquad (7)$$

The formulation in (7) is a variant of the popular Wasserstein-GAN (Arjovsky et al., 2017), except that the noise $\mathbf{z}$ is conditioned on the input sequence, and that the input data $\mathbf{x}$ is projected into some subspace $\mathbf{U}$ to be learned. Note that while it may seem we can optimize all the parameters together alongside learning the representation $\mathbf{U}$ using (7), it poses some technical hurdles. For example, unless we know how to generate the negative distribution via learning θ, we cannot produce the representation $\mathbf{U}$, and without this representation, we cannot use $f_{\mathbf{U}}$. This poses a challenge with learning the two jointly, especially when considering highly non-linear neural networks to characterize h. A second problem is that the data from a single sequence may be insufficient to learn θ. We circumvent both these problems by separating the representation learning objective from the noise generation objective; the latter we re-formulate as:

$$\mathcal{L}_G(\theta) := \min_{\theta} \max_{h\in L_1} \mathbb{E}_{\mathbf{x}\in\mathbf{X}\sim\mathcal{D}}\left[h(\mathbf{x})\right] - \mathbb{E}_{\mathbf{y}\sim g_\theta(\overline{\mathbf{X}},\sigma)}\left[h(\mathbf{y})\right], \qquad (8)$$

where now, we removed $f_{\mathbf{U}}$ and included the parameter learning problem using the full dataset $\mathcal{D}$, instead of a single sequence $\mathbf{X}$.

3.3. Adversarial Noise Generation

While, our formulation in (8) does allow learning noise distributions that are similar in distribution to the data samples $\mathbf{x}$, it is not adversarial in the sense that there is nothing preventing the generated noise from being adequately discriminative, i.e., from mimicking spatio-temporal features; for example, $\mathbf{Y}$ may have features that are useful for recognition, however we desire our final representation to be maximally different from this noise. To account for these requirements, we propose to learn to generate adversarial noise. We need some new notation to present our technique.

Suppose we have a pre-trained (frame-level) classifier $\zeta : \mathbb{R}^d \to |\mathcal{L}|$ that takes each $\mathbf{x} \in \mathbf{X}$ and returns a class label $\ell_{\mathbf{X}}$ associated with $\mathbf{X}$. Different from (7), now our objective is not to just produce noise, but to make this noise adversarial to the useful components in the input data. That is, for a given feature $\mathbf{x} \in \mathbf{X}$, we seek to generate a noise sample $\hat{\mathbf{x}}$ from $g_\theta(\overline{\mathbf{X}}, \sigma)$ such that when this sample is subtracted from the input $\mathbf{x}$, a classifier $\zeta(\mathbf{x})$ that is trained to produce $\ell_{\mathbf{X}}$ will misclassify; i.e., if $\zeta(\mathbf{x}) \to \ell_{\mathbf{X}}$, then $\zeta(\gamma(\mathbf{x} - \hat{\mathbf{x}}))|\hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \sigma) \to \bar{\ell}_{\mathbf{X}}$, where $\bar{\ell}_{\mathbf{X}} \in \mathcal{L}$ is any other class label[3] other than $\ell_{\mathbf{X}}$, and γ is a suitable operator; e.g., a ReLU ensuring $\gamma(\mathbf{x} - \hat{\mathbf{x}})$ remains in the same feature space as $\mathbf{x}$. Incorporating these requirements into our GAN objective

[3] In practice, WGAN usually learns to generate random noise of arbitrary strength that could misclassify the input, without accounting for useful subspaces. To circumvent this issue, our objective demands that the generated noise when combined with the input data will ensure the useful data properties are removed. We do this by asking the classifier to produce a logit vector such that its softmin is equal to ℓ, while for the non-perturbed input, we ask the softmax be equal to ℓ.

in (7), we have our new adversarial WGAN objective as:

$$\mathcal{L}_A(\theta) = \min_{\theta} \max_{h \in L_1} \mathbb{E}_{\mathbf{x} \in \mathbf{X} \sim \mathcal{D}} \left[h(\mathbf{x})\right] - \mathbb{E}_{\substack{\mathbf{y}=\gamma(\mathbf{x}-\hat{\mathbf{x}}), \\ \hat{\mathbf{x}} \sim g_\theta(\overline{\mathbf{X}}, \sigma)}} \left[h(\mathbf{y})\right] + \lambda_1 \left(\zeta(\mathbf{x}, \ell_{\mathbf{X}}) - \zeta(\gamma(\mathbf{x}-\hat{\mathbf{x}}), \ell_{\mathbf{X}})\right) + \lambda_2 \mathbb{E}[\|\hat{\mathbf{x}}\|^2]. \quad (9)$$

The regularization $\mathbb{E}[\|\hat{\mathbf{x}}\|^2]$ is useful to make g_θ learn to produce noise of small strength that can misclassify the input (similar to standard adversarial models (Moosavi-Dezfooli et al., 2016)). The positive weights λ's balance the losses at training. Note that we need to input $\overline{\mathbf{X}}$ to the generator g_θ so that the generator learns to know the noisy features in its input (at the frame level). Our full objective in (9) is trained end-to-end for the WGAN.

3.4. Capturing Sequentiality and Distortion

Now that, we have a concrete setup to sample from adversarial noise distributions to generate the negative samples, lets look back at the representation learning objective in (4). Recall that using the noise distributions, we have accounted for finding samples $\mathbf{Y}$ that do not have *useful* data properties that were present in the input data $\mathbf{X}$. While, using $\mathbf{Y}$ in OT allows for a high and relevant discrimination from $\mathbf{X}$; however, the constrastive representation should also account for the sequentiality in the data. To this end, we include temporal ordering constraints on the learned subspaces. Specifically, we ask the subspace to be learned in such a way that when data points is projected onto this subspace, some monotonicity property is satisfied. That is, for some $\eta > 0$, the projections of each $\mathbf{x}_t$ onto the learned subspace $\mathbf{U}$ satisfies ordering constraints: $\left\|\mathbf{U}^\top \mathbf{x}_t\right\|^2 + \eta \leq \left\|\mathbf{U}^\top \mathbf{x}_{t+1}\right\|^2, \forall t = 1, 2, \cdots, n-1$. We also ensure that the representation does not diverge too much from the input sequence, via including a distortion penalty into our objective. Note that these constraints have been used previously, such as in (Cherian et al., 2017). With these additional constraints, we rewrite our full representation learning objective in (4) as:

$$\max_{\mathbf{U} \in \mathcal{G}(d,k)} \mathcal{L}_R(\mathbf{U}) := \mathcal{L}_{OT}(\mathbf{U}) - \frac{\beta_1}{n} \sum_{\mathbf{x} \in \mathbf{X}} \|f_{\mathbf{U}}(\mathbf{x}) - \mathbf{x}\|^2 - \frac{\beta_2}{n-1} \sum_{t=1}^{n-1} \left[\left\|\mathbf{U}^\top \mathbf{x}_t\right\|^2 + \eta - \left\|\mathbf{U}^\top \mathbf{x}_{t+1}\right\|^2\right]_+, \quad (10)$$

where the βs are constants, and $[\,.\,]_+ = \max(0, .)$.

3.5. Representation Learning

For a given sequence $\mathbf{X} \in \mathbb{R}^{d \times n}$ and its adverserially sampled noise matrix $\mathbf{Y} \in \mathbb{R}^{d \times m}$, we can rewrite our full objective in (10) as:

$$\max_{\mathbf{U} \in \mathcal{G}(d,k)} \min_{\pi \in \Delta(n,m)} \langle \pi, \mathrm{dist}(f_{\mathbf{U}}(\mathbf{X}), \mathbf{Y}) \rangle - \Omega(\mathbf{X}, \mathbf{U}), \quad (11)$$

where $\Delta(n,m)$ is a set of linear constraints capturing the marginals, and dist produces an $n \times m$ distance matrix. Further, with a slight abuse of notation, we assume $f_{\mathbf{U}}(\mathbf{X}) = \mathbf{U}\mathbf{U}^\top \mathbf{X}$, and $\Omega(\mathbf{X}, \mathbf{U})$ captures the distortion and the ordering constraints on $\mathbf{U}$. We propose to use alternating minimization to solve this objective, where we alternatingly solve the following sub-problems, while keeping the other optimization variable fixed. Specifically, by fixing $\mathbf{U}$ (initially assuming $\mathbf{U}\mathbf{U}^\top = I$), we solve for π as:

$$\min_{\pi \in \Delta(n \times m)} \langle \pi, \mathrm{dist}(f_{\mathbf{U}}(\mathbf{X}, \mathbf{Y}) \rangle, \quad (12)$$

which we solve efficiently using an inexact proximal point optimal transport (IPOT) solver proposed in (Xie et al., 2019). In contrast to entropy regularization based methods (Cuturi, 2013), IPOT has better convergence and stability properties. Fixing the coupling π, we solve for the subspace $\mathbf{U}$ via casting the optimization on the Grassmann manifold. That is, we solve:

$$\min_{\mathbf{U} \in \mathcal{G}(d,k)} - \left\|\mathbf{U}\mathbf{U}^\top \mathbf{X} - \mathbf{Y}\pi^\top\right\|_F^2 + \Omega(\mathbf{X}, \mathbf{U}). \quad (13)$$

We use the Riemannian conjugate gradient algorithm (Absil et al., 2009) to optimize this sub-problem. We use the Fletcher and Reeves step size selection (Fletcher & Reeves, 1964) and retractions via the QR decomposition for the iterations. As each of our sub-problems is non-convex, it is not easy to guarantee any convergence. However, empirically, we see that the IPOT solver finds the coupling in about 1000 iterations (note that each iteration is very cheap) and the Riemannian conjugate gradient converges in about 5 iterations.

3.6. Representation Classifier

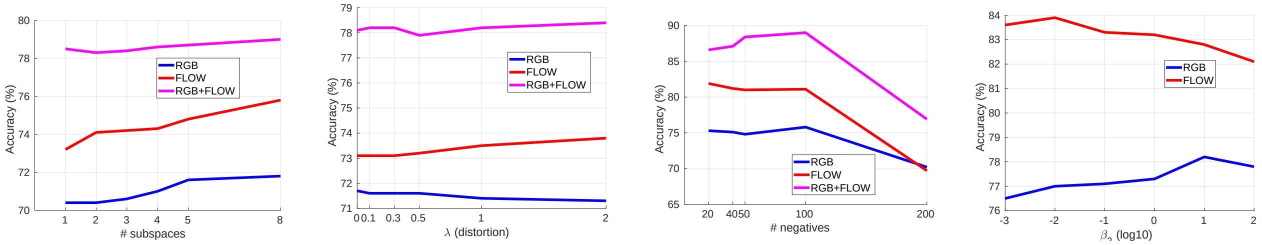
The missing piece to present in our framework is the classifier C to use on the subspace representations, defined in (1). A tricky problem with subspaces is that there is no control on the sign of their basis. While, we may use a deep neural network classifier to this end, in this work, we use a standard kernelized SVM, with a kernel $\mathbf{K}$ defined for two points $\mathbf{U}_1, \mathbf{U}_2 \in \mathcal{G}(d,k)$ as (Harandi et al., 2014):

$$\mathbf{K}(\mathbf{U}_1, \mathbf{U}_2) = \exp\left(\gamma \left\|\mathbf{U}_1^\top \mathbf{U}_2\right\|_F^2\right), \quad (14)$$

for a bandwidth parameter $\gamma > 0$. A computationally cheaper alternative to using a kernel, which we found to produce similar results empirically, is to use average pooling after computing $\mathbf{U}\mathbf{U}^\top \mathbf{X}$. This results in a representation in $\mathbb{R}^d$, and is especially desirable if using a linear sequence classifier.

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**	**73.2**	**75.5**	**79.4**

Table 1. Ablative Study of various modules in our framework and their benefits on the JHMDB dataset (with two different types of frame-level features, i.e.,VGG and I3D) and the HMDB dataset. We report performances on these datasets for the RGB stream, the optical flow stream, and their combination in a two-stream action recognition setup. The results are based on the split-1 of the respective datasets. The acronyns are as follows: A=Adversarial, C=Contrastive, OT=optimal transport, PCA=principal components analysis, and Random=using random noise instead of adversarially generated noise, order=Temporal ordering.



(a) Increasing subspaces k (b) Distortion regularization β_1 (c) # Negatives (d) Temporal regularization β_2

Figure 2. Sensitivity analysis of various choices in our representation learning setup. See text for details. Left two plots are on the HMDB dataset, the right two are on the JHMDB dataset.

4. Experiments

In this section, we present experiments demonstrating the effectiveness of our approach for representation learning. To this end, we use two standard action recognition datasets, namely (i) small-scale JHMDB (Jhuang et al., 2013) and (ii) the larger HMDB dataset (Kuehne et al., 2011). We also present some qualitative results on the CIFAR dataset. More experiments and results are provided in the supplementary materials, due to lack of space.

JHMDB dataset: consists of 928 video sequences, each sequence about 15-40 frames long. There are 21 actions defined on the clips, and each clip has only one action. We use this dataset as a test-bed to explore the performances of various modules in our setup. To this end, we use a standard two stream neural network (Simonyan & Zisserman, 2014) for extracting video features at the frame-level and from a short-clip of 20 optical flow frames. To make our results comparable to prior works, we use vgg-16 features made available as part of (Cherian et al., 2017). We also evaluate using 3D CNN features as produced by a pre-trained I3D network (Carreira & Zisserman, 2017) in a two-stream setup.

HMDB Dataset: is a super-set of the JHMDB dataset and consists of about 6700 video clips, each with 50–400 frames and 51 actions. We use a pre-trained I3D network (trained on Kinetics) in a two-stream framework for evaluating the performance of our model on this dataset.

Hyperparameters: Our entire implementation is in PyTorch. For our adversarial module, we modified the public WGAN code associated with (Arjovsky et al., 2017). We used a noise variance $\sigma = 0.01$, which resulted in an average classifier fooling rate of 60% on the training set on both the datasets. See the Appendix for more experiments in this regard. Further, we used $\lambda_1 = 0.1$ and $\lambda_2 = 1$ in (9). We used PyManOpt as our Riemannian optimization framework[4]. As for the regularization constants on the distortion and ordering constraints in (10), we set $\beta_1 = 1$ and $\beta_2 = 10$, and we used $\eta = 0.01$ for the temporal margin. We also assume that all features from the two datasets are normalized to unit ℓ_2 norm. We will be making our code publicly available at `https://www.merl.com/research/license/`.

4.1. Ablative Studies

In Table 1, we provide a thorough ablative study of the various modules in our framework. Specifically, we compare (i)

[4] `https://www.pymanopt.org/`

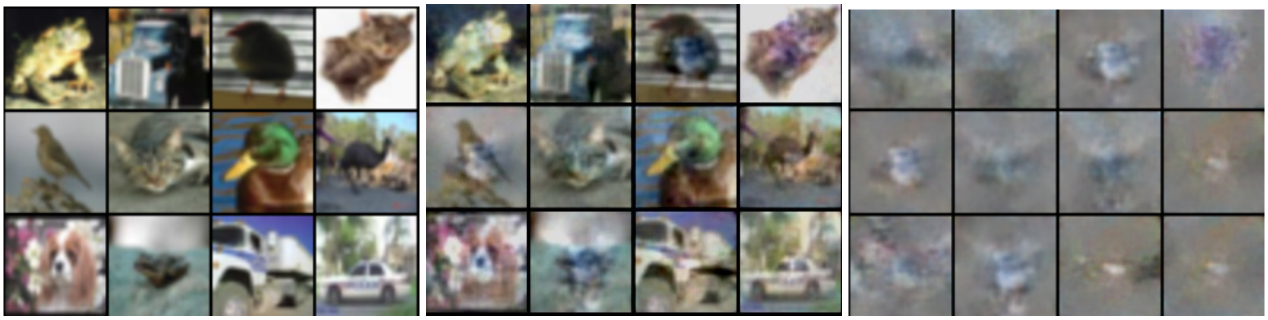
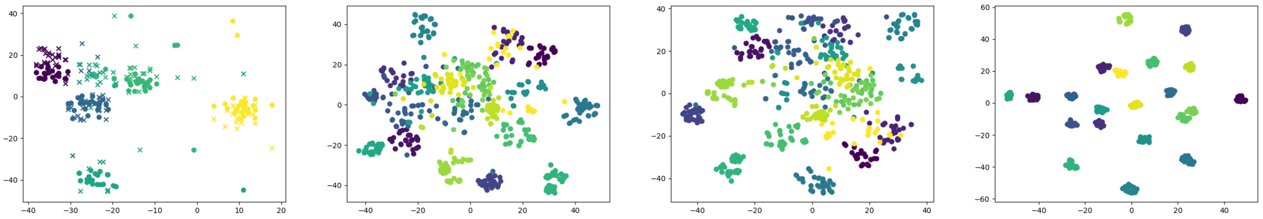


Figure 3. Qualitative results showing (left) the original CIFAR image, (middle) image added with sampled noise from an adversarial distribution, and (right) the respective sampled noise. See text for details.



(a) Adv. Samples

(b) Average Pooling

(c) Random Contrastive

(d) Adv. Contrastive

Figure 4. T-SNE plots on the JHMDB dataset using I3D features. Figure 4(a) shows features (circles) and their respective adversarial negative samples as 'x's (shown for 5 classes). Each color corresponds to a distinct class. Note that the negative samples ('x's) for each class are very close in embedding to the (positive) data samples (circles). In Figure 4(b), we plot the embeddings of sequence representations produced via average pooling of frame-level features (for all 21 classes). In Figure 4(c), we use our contrastive optimal transport for representation learning, however uses random noise to contrast against. In Figure 4(d), we embed the representations produced via adversarially-constrastive OT (ACOT). Our scheme results in well-separated clusters, justifying the superior performances.

average pooling, which provides the baseline performance on the dataset, (ii) Contrastive Optimal Transport (COT) without the adversarial noise, instead using Gaussian noise (iii) adversarially contrastive optimal transport (ACOT), (iv) ACOT with the distortion penalty (PCA) as characterized via a data reconstruction loss, (v) adversarially contrastive estimation with the PCA and temporal ordering constraints, but without the optimal transport cost, and (vi) our full model (ACOT+PCA+temporal order). For (ii), we found that using random noise that is completely uncorrelated with the data resulted in very poor performance (COT+Random). For this experiment, we produced the negative samples by multiplying random noise with the maximum feature value and subtracting it from the feature. That is, for a feature $\mathbf{x}$, we generate $\mathbf{z} \sim \mathcal{N}(0, \max(\mathbf{x}))$, and compute the negative examples $\mathbf{y} = \gamma(\mathbf{x} - \mathbf{z})$. Here, $\max(\mathbf{x})$ computes the maximum value in the dimensions of $\mathbf{x}$. We repeated this ablative study on the two datasets over vgg and I3D features.

As is clear from the Table 1, using COT leads to better performances than average pooling in most cases, with ACOT demonstrating better performance than COT+Random in most cases. We found that there is a significant variance ($\pm 2\%$) on COT+Random and thus our numbers are averaged over 5 trials. For ACOT, we sampled twice the number of negative samples as the positives. The combinations of PCA and order also demonstrates benefits overall, showing the effectiveness of our proposed method in extracting weak temporal signals from the sequential data. On the JHMDB dataset, our full architecture is significantly better than average pooling by nearly 7% on both vgg and I3D features. A similar observation is made on the HMDB dataset as well. Our full model is about 2-5% better on the RGB and FLOW streams separately and about 2% better in combination. This clearly demonstrates the generalizability of our method to different datasets, types of features, and data modalities (RGB and optical flow).

Increasing Number of Subspaces: In Figure 2(a), we keep the distortion $\beta_1 = 0.1$ and number of negatives to 50, and plot the performance of ACOT against increasing number of subspaces in $\mathbf{U}$. As we see, there is an increasing trend in the performance (on the HMDB dataset). However, as the number of subspaces increases, the representation learning time also increases (about k times slower for k subspaces). Beyond 8 subspaces, we did not find any improvements.

Increasing Distortion Penalty: Keeping $\#$-subspaces at

1, and #-negatives at 50, we increased β_1 from 0 to 2. As seen in Figure 2(b), we see a marginal improvement in performance with increasing β_1. We believe, the contrastive formulation is already capturing useful properties of the input sequences that the contribution from an explicit distortion penalty is incremental.

Increasing Number of Negatives: An advantage of our setup is the possibility of generating unlimited number of negatives. In Figure 2(c), we increased the number of negative samples from 20 to 200. While, it is very expensive for the OT to solve for large negative sets, we found that using about 40-100 samples is adequate and demonstrates performance improvements. However, with a large number of negatives (such as 200), counter-intuitively, we find that the accuracy drops significantly. This is perhaps because our noise generator model does not fool the classifier perfectly; as a result, overabundant negatives may be overlapping in distribution to the positives; diminishing the contrastiveness.

Increasing Temporal Ranking Regularization: In this experiment, we kept $\beta_1 = 0.1$, number of negatives to 50, and number of subspaces to 1, and changed the temporal regularization β_2 (in (10)). Figure 2(d) plots this sensitivity. As is clear from the figure, smaller regularization is ineffective.

Running Time: On average, excluding the time to extract CNN features, our representation learning setup takes about 30 frames per second using 5 iterations of conjugate gradient and 1000 iterations of inexact proximal optimal transport.

4.2. Comparisons to the State of the Art

In Tables 2 and 3, we compare the performances of our method against prior works on the respective datasets, such as (i) GRP (Cherian et al., 2017) which learns order-constrained subspaces for sequence data, similar to ours, (ii) KRP (Cherian et al., 2018) that extends GRP to use kernels, and P-CNN (Chéron et al., 2015) which is a baseline that uses vgg features. Against these baselines, our method performs better by about 2% on three-fold cross-validation. We repeated this experiment with I3D features, and compare to a few recent methods that also use 3D convolutions. Again, our full model is seen to improve the state-of-the-art by nearly 2% on this dataset. Note that we did not train our I3D model on the dataset, instead passed the frames through a pre-trained network. To this end, for evaluations on the HMDB dataset, we applied the code for GRP (Cherian et al., 2017) on our features, while we used random noise for Discriminative Pooling (Wang & Cherian, 2019).

4.3. Qualitative Visualizations

To gain insights into the kind of perturbations our adversarial network generates, we trained this sub-module on the CIFAR10 dataset. Specifically, we use an auto-encoder on the CIFAR images, the latent vectors of each image forms the feature. Next, we trained our adversarial network, similar to our setup for sequences, but assuming only a single frame (the encoded CIFAR image). We combined the generated noise with the latent feature and decoded the image. The goal of the discriminator in our framework is to classify this generated image as "fake", while the generator is trained to produce better noise such that the discriminator is fooled, however, a pre-trained CIFAR classifier (trained on the 10 CIFAR classes) shows a low-confidence against the true image class (i.e., the noise needs to be adversarial). In Figure 3, we show a few qualitative CIFAR images. As is clear from the images, the noise impacts regions of the image that are likely to be useful for recognition (for example, the image of the cat on the top-right, and its respective perturbation). This experiment illustrates that our generated noise masks the useful data subspaces. In Figure 4, we show T-SNE embeddings of representations learned on JHMDB dataset features.

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**

Table 2. Comparisons on JHMDB dataset (3-splits).

Method	Acc. (%)
Two Stream (Simonyan & Zisserman, 2014)	65.9
3D-fused (Carreira & Zisserman, 2017)	64.6
I3D (Carreira & Zisserman, 2017)	78.6
GRP (Cherian et al., 2017)	77.6
Disc. Pool (Wang & Cherian, 2019)	78.0
Ours (I3D+full model)	**79.4**

Table 3. Comparisons on HMDB dataset (split 1).

5. Conclusions

We presented a novel framework for producing data representations on sequences via contrastive learning. Our key insight to look at this problem emerged from the observation that each item in a (video) sequence is often encoded using a model that does not access the full sequence. As a result, the cues for sequence level inference within such encodings might be weak. To amplify such cues, we resorted to contrastive learning, where we contrast the features against adversarially-learned features, and learns subspaces, as a representation, that captures these weak cues via optimal transport. We presented experiments on two datasets, demonstrating empirical benefits against recent methods.

A. Proof of Theorem 1

Proof. Here we will prove a slightly general form of Theorem 1. We begin by noting that,

$$\mathcal{P}_k^2 \triangleq \max_{\mathbf{U}:\mathcal{S}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \mathbb{E}_\pi \|\mathbf{U}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{y}\|^2,$$
$$= \max_{\mathbf{U}:\mathcal{G}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y})})} \mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{y}\|^2$$

and

$$\mathcal{S}_k^2 \triangleq \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \max_{\mathbf{U}\in\mathcal{S}(d,k)} \mathbb{E}_\pi \|\mathbf{U}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{y}\|^2.$$
$$= \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{y}\|^2.$$

Now,

$$\begin{aligned}
\mathcal{C}_k^2 &= \max_{\mathbf{U}\in\mathcal{G}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{y}\|^2 \\
&= \max_{\mathbf{U}\in\mathcal{G}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \mathbb{E}_\pi \{\|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{y}\|^2 + \\
&\qquad \|(\mathbf{I}_d - \mathbf{U}\mathbf{U}^\top)\mathbf{y}\|^2\} \\
&\geq \max_{\mathbf{U}\in\mathcal{G}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{y}\|^2 \qquad (15) \\
&= \mathcal{P}_k^2 \qquad (16)
\end{aligned}$$

Now since $\max\min \leq \min\max$,

$$\begin{aligned}
&\max_{\mathbf{U}\in\mathcal{G}(d,k)} \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x} - \mathbf{y}\|^2 \\
&\leq \min_{\pi\in\Pi(\mu_{\mathbf{X}},\nu_{\mathbf{Y}})} \max_{\mathbf{U}\in\mathcal{G}(d,k)} \{\mathbb{E}_\pi \|\mathbf{U}\mathbf{U}^\top \mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{y}_j\|^2 + \\
&\qquad \mathbb{E}_\pi \|(\mathbf{I}_d - \mathbf{U}\mathbf{U}^\top)\mathbf{y}\|^2\} \\
&\leq \mathcal{S}_k^2 + \max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathbb{E}_{\nu_{\mathbf{Y}}} \|(\mathbf{I}_d - \mathbf{U}\mathbf{U}^\top)\mathbf{y}\|^2. \qquad (17)
\end{aligned}$$

The term $\max_{\mathbf{U}\in\mathcal{G}(d,k)} \mathbb{E}_{\nu_{\mathbf{Y}}} \|(\mathbf{I}_d - \mathbf{U}\mathbf{U}^\top)\mathbf{y}\|^2 = \sum_{\ell=k+1}^d e_\ell(\mathbf{\Sigma}_{\mathbf{Y}})$ where $e_1, e_2, ..., e_d$ are the eigenvalues of the Gram matrix arranged in increasing order. □

B. Additional Experiments

In this section, we detail our neural architectures in our COT framework and provide ablative studies of the various choices in our setup.

Datasets and Features: As noted in the main paper, we use two datasets, namely (i) the JHMDB dataset, and (ii) the HMDB dataset. For the former, we explore our scheme using two types of features: (i) vgg-16 features, and (ii) I3D features. The vgg-16 features are 4096 dimensional each for every frame in the sequence. That is, we have feature matrices of size $4096 \times n$ and $4096 \times n - 1$ for the RGB and optical flow respectively, where n denotes the number of frames in the sequence. As for the I3D features, they are 1024 dimensional each and are extracted from the average pooling layer (after the "max_5c" layer) of the Inception V3 network (Carreira & Zisserman, 2017). These features are produced from short clips, in which the I3D network takes clips consisting of 8 consecutive video frames, and produces one 1024 dimensional feature for that short clip. We use a sliding window with a temporal stride of 2 frames to generate our feature matrix for the two streams. Thus, in our setup, for a sequence with n frames, we will have feature matrices of size $1024 \times \lfloor \frac{n}{2} \rfloor$ and $1024 \times \lfloor \frac{n-1}{2} \rfloor$ for the RGB and flow streams respectively. Note that the features (from either network) are the outputs of ReLU activations and thus are all non-negative. We also normalize these features to have unit-norm.

Baseline Networks and Training: As alluded to in the main paper, we have not trained the baseline networks ourselves as our goal is to demonstrate the advantages of adversarially constrastive optimal transport on features extracted from off-the-shelf neural models. To this end, for the vgg-16 features on the JHMDB dataset, we directly use the features provided to us by the authors of (Cherian et al., 2017). As is mentioned in that paper, these features were infact produced using a network that was fine-tuned on the JHMDB dataset. For the I3D features, we used a ImageNet+Kinetics pretrained I3D network implemented in PyTorch from a public git-hub repo[5] to extract the features as described above.

B.1. Neural architectures

Apart from the baseline feature-generating neural networks as described above, our framework has two other neural sub-modules, namely (i) the Wasserstein GAN (WGAN) framework for generating the adversarial samples, and (ii) the classifier to ensure the samples are adversarial.

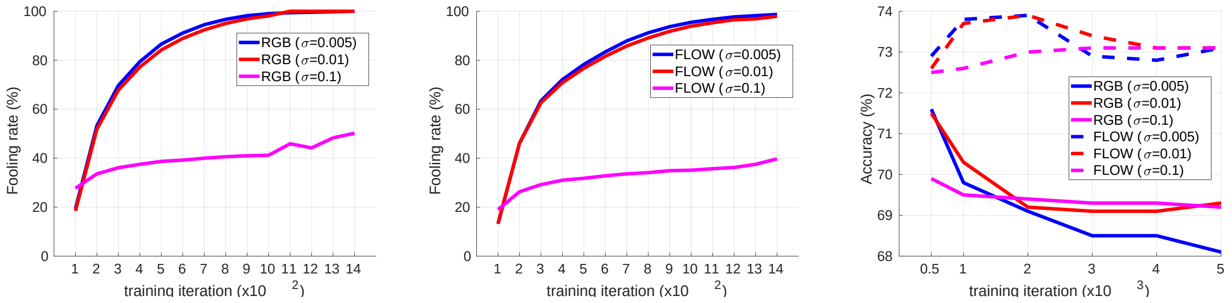
Generator and Discriminator: Our generator g has the following neural composition:

$$g := [\text{FCN}(d, d), \text{ReLU}(), \text{FCN}(d, d), \text{ReLU}(), \text{FCN}(d, d)]$$

where d is the input feature dimensionality (4096 for vgg-16 and 1024 for I3D), where this input is a noise sample from a multivariate normal distribution. Our discriminator has a similar structure, except that the final layer uses $\text{FCN}(d, 1)$.

Classifier: As our representations for the sequences are linear subspaces, we decided to have the adversarial classifier also be limited in capacity, and thus we used a linear classifier for ζ in (10). Specifically, our classifier consists of a single $\text{FCN}(d, c)$, where c denotes the number of data classes. We attempted adding more layers and non-linearities to this classifier, however we found that such attempts made it difficult for the generator to learn the perturbations, and also the learned perturbations were difficult to be separated using the linear subspaces U in our ACOT scheme.

[5] https://github.com/piergiaj/pytorch-i3d



(a) RGB (b) FLOW (c) ACOT Accuracy

Figure 5. Fooling rates for RGB stream (Figure 5(a)) and FLOW stream (Figure 5(b)) using I3D network on the HMDB dataset against the number of WGAN training iterations. We plot for three different variances of the Normal distribution, i.e., $\sigma = 0.005, 0.01, 0.1$. Note that standard deviation of the features is about 0.008. As we see from the two plots, with a lower $\sigma = 0.005, 0.01$, the WGAN learns to generate adverarial pertubations with 100% fooling rate in about 1000 iterations, however with a larger $\sigma = 0.1$, the network could achieve about 50% fooling rate on average. On the right 5(c), we plot the validation accuracy (of ACOT) against the respective training iterations on the left. For RGB, higher-fooling rates seem to affect performances, however, the effect is reversed on the FLOW stream. This is perhaps because the RGB stream of I3D does not capture any useful temporal cues.

B.2. Adversarial Training

We used RMSprop for training our models. We used a learning rate of $1e-4$ for the generator and discriminator, and for the classifier. We trained the classifier for 500 iterations and it achieves roughly 80% accuracy on the input features (on the training set). More training resulted in overfitting, and thus posed difficulties when training the subsequent adversarial network. For WGAN, we adapted the public implementation from the authors of (Arjovsky et al., 2017). This code uses 5 discriminative updates for every generator updates, which we also found to be useful in our setup. We measured the quality of the generated perturbations via the fooling rates on the positive samples. Specifically, the generated random perturbations are added to the original data samples (positives), passed through a $\mathrm{ReLU}()$, and then normalized to unit norm (note that all our data is unit-normalized) to produce the negative samples. Thus, if c is the correct class label that a classifier ζ produces on an input $\mathbf{x}$, then $\mathbf{y} = \frac{\mathrm{ReLU}(\mathbf{x}+g(\mathbf{z}))}{\|\mathrm{ReLU}(\mathbf{x}+g(\mathbf{z}))\|}$, where $\mathbf{z} \sim \mathcal{N}(\overline{\mathbf{X}}, \sigma^2 I)$ is classified as $\bar{c}$ by ζ, where $\bar{c}$ means the class c has the lowest likelihood of being predicted, i.e., $c = \mathrm{softmin}(\zeta(\mathbf{y}))$. We define fooling rate as the performance of the generator to produce a $\mathbf{y}$ that fools ζ as described. Figure 5 show the trend in training the WGAN for various choices of σ and its impact on the ACOT performance. Please see the text accompanying Figure 5 for the empirical analysis. Going by that analysis, we use $\sigma = 0.01$ in our experiments.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Bose, A. J., Ling, H., and Cao, Y. Adversarial contrastive estimation. *arXiv preprint arXiv:1805.03642*, 2018.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Cherian, A., Fernando, B., Harandi, M., and Gould, S. Generalized rank pooling for activity recognition. In *CVPR*, 2017.

Cherian, A., Sra, S., Gould, S., and Hartley, R. Non-linear temporal subspace representations for activity recognition. In *CVPR*, 2018.

Chéron, G., Laptev, I., and Schmid, C. P-CNN: Pose-based CNN features for action recognition. In *ICCV*, 2015.

Choutas, V., Weinzaepfel, P., Revaud, J., and Schmid, C. Potion: Pose motion representation for action recognition. In *CVPR*, 2018.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

Cover, T. and Thomas, J. *Elements of Information Theory*. John Wiley and Sons, 2006.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. Behavior recognition via sparse spatio-temporal features. In *Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

Fletcher, R. and Reeves, C. M. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.

Girdhar, R., Tran, D., Torresani, L., and Ramanan, D. Distinit: Learning video representations without a single labeled video. *CoRR*, abs/1901.09244, 2019. URL http://arxiv.org/abs/1901.09244.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2016.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

Harandi, M. T., Salzmann, M., Jayasumana, S., Hartley, R., and Li, H. Expanding the family of grassmannian kernels: An embedding perspective. In *ECCV*, 2014.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S. M. A., and van den Oord, A. Data-efficient image recognition with contrastive predictive coding. *CoRR*, abs/1905.09272, 2019. URL http://arxiv.org/abs/1905.09272.

Hoffer, E. and Ailon, N. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2015.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. Towards understanding action recognition. In *ICCV*, 2013.

Kim, T., Ahn, S., and Bengio, Y. Variational temporal abstraction. In *NeurIPS*. 2019.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

Merity, S., Keskar, N. S., and Socher, R. Regularizing and optimizing lstm language models. In *ICLR*, 2018.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paty, F. and Cuturi, M. Subspace robust wasserstein distances. *CoRR*, abs/1901.08949, 2019. URL http://arxiv.org/abs/1901.08949.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *ICML*, 2019.

Santambrogio, F. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, 2019.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

Smith, N. A. and Eisner, J. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*, 2005.

Song, J. and Ermon, S. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

Sun, C., Baradel, F., Murphy, K., and Schmid, C. Contrastive bidirectional transformer for temporal representation learning. *CoRR*, abs/1906.05743, 2019. URL http://arxiv.org/abs/1906.05743.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Tran, D., Ray, J., Shou, Z., Chang, S., and Paluri, M. Convnet architecture search for spatiotemporal feature learning. *CoRR*, abs/1708.05038, 2017. URL http://arxiv.org/abs/1708.05038.

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Varol, G., Laptev, I., and Schmid, C. Long-term temporal convolutions for action recognition. *CoRR*, abs/1604.04494, 2016.

Varol, G., Laptev, I., and Schmid, C. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (6):1510–1517, 2017.

Wang, J. and Cherian, A. Learning discriminative video representations using adversarial perturbations. *CoRR*, abs/1807.09380, 2018. URL http://arxiv.org/abs/1807.09380.

Wang, J. and Cherian, A. Discriminative video representation learning using support vector classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019. doi: 10.1109/TPAMI.2019.2937292.

Wang, X. and Gupta, A. Unsupervised learning of visual representations using videos. *CoRR*, abs/1505.00687, 2015.

Wu, C., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., and Krähenbühl, P. Compressed video action recognition. *CoRR*, abs/1712.00636, 2017.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing exact wasserstein distance. In *UAI*, 2019.

Zilly, J. G., Srivastava, R. K., Koutnık, J., and Schmidhuber, J. Recurrent highway networks. In *ICML*, 2017.

Zolfaghari, M., Oliveira, G. L., Sedaghat, N., and Brox, T. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*, 2017.