Hierarchical Relational Networks for Group Activity Recognition and Retrieval

Mostafa S. Ibrahim and Greg Mori

School of Computing Science, Simon Fraser University, Canada
msibrahi@sfu.ca, mori@cs.sfu.ca

Abstract. Modeling structured relationships between people in a scene is an important step toward visual understanding. We present a Hierarchical Relational Network that computes relational representations of people, given graph structures describing potential interactions. Each relational layer is fed individual person representations and a potential relationship graph. Relational representations of each person are created based on their connections in this particular graph. We demonstrate the efficacy of this model by applying it in both supervised and unsupervised learning paradigms. First, given a video sequence of people doing a collective activity, the relational scene representation is utilized for multi-person activity recognition. Second, we propose a Relational Autoencoder model for unsupervised learning of features for action and scene retrieval. Finally, a Denoising Autoencoder variant is presented to infer missing people in the scene from their context. Empirical results demonstrate that this approach learns relational feature representations that can effectively discriminate person and group activity classes.

1 Introduction

Human activity recognition is a challenging computer vision problem and has received a lot of attention from the research community. Challenges include factors such as the variability within action classes, background clutter, and similarity between different action classes. Group activity recognition arises in the context of multi-person scenes, including in video surveillance, sports analytics, and video search and retrieval. A particular challenge of group activity recognition is the fact that inferring labels for a scene requires contextual reasoning about the people in the scene and their relations. In this paper we develop a novel deep network layer for learning representations for capturing these relations.

Fig. 1 provides a schematic of our relational layer and Fig. 2 highlights the processing of a single person inside the layer. Initially, each person in a scene can be represented by a feature, e.g. derived from a standard CNN. We amalgamate these individual representations via stacking multiple relational layers – deep network layers that combine information from a set of (neighbouring) person representations. These layers are utilized in a hierarchy, refining representations for each individual person based on successive integration of information from other people present in the scene.

Recent deep learning approaches [9, 20, 25] for group activity recognition use a 2-stage processing pipeline where first each person is represented using a large feature

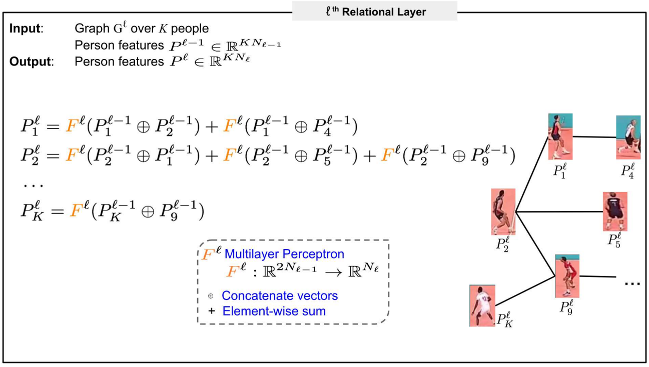

ℓ^{th} Relational Layer
Input: Graph G^ℓ over K people
Person features $P^{\ell-1} \in \mathbb{R}^{KN_{\ell-1}}$
Output: Person features $P^\ell \in \mathbb{R}^{KN_\ell}$
$P_1^\ell = F^\ell(P_1^{\ell-1} \oplus P_2^{\ell-1}) + F^\ell(P_1^{\ell-1} \oplus P_4^{\ell-1})$
$P_2^\ell = F^\ell(P_2^{\ell-1} \oplus P_1^{\ell-1}) + F^\ell(P_2^{\ell-1} \oplus P_5^{\ell-1}) + F^\ell(P_2^{\ell-1} \oplus P_9^{\ell-1})$
$\dots$
$P_K^\ell = F^\ell(P_K^{\ell-1} \oplus P_9^{\ell-1})$
F^ℓ Multilayer Perceptron
$F^\ell : \mathbb{R}^{2N_{\ell-1}} \rightarrow \mathbb{R}^{N_\ell}$
$\oplus$ Concatenate vectors
+ Element-wise sum
P_1^ℓ
P_4^ℓ
P_2^ℓ
P_5^ℓ
P_9^ℓ
...
P_K^ℓ

Fig. 1: A single relational layer. The layer can process an arbitrary sized set of people from a scene, and produces new representations for these people that capture their relationships. The input to the layer is a set of K people and a graph G^ℓ encoding their relations. In the relational layer, a shared neural network (F^ℓ) maps each pair of person representations to a new representation that also encodes relationships between them. These are aggregated over all edges emanating from a person node via summation. This process results in a new, relational representation for each of the K people. By stacking multiple relational layers, each with its own relationship graph G^ℓ, we can encode hierarchical relationships for each person and learn a scene representation suitable for group activity recognition or retrieval.

vector (e.g., fc7 features). Then, the person representations are pooled together to construct the final features for the scene. The typical scene pooling is max / average / attentional pooling over people, which reduces dimensionality, but loses information. First, all spatial and relational information is dropped. Second, features about individual people, which actually define actions, are lost. Finally, although such a scene representation is optimized for group activity recognition, it cannot be used for analysis tasks based on individual *actions*.

Our models utilize a similar 2-stage processing framework, but work on solving these drawbacks in an efficient and effective manner. Given initial feature representations for each person and a relationship graph, we present a relational layer that jointly computes a compact representation for each person that encodes inter-person relations. By stacking multiple relational layers, this hierarchical relational network learns a *compact relational representation per person.*

Our contributions can be summarized as follows:

- A relational layer that jointly infers relational representations for each person based on a relationship graph. The layer can operate on a variable sized set of people in a scene. Given features for K people, the layer maps the given K feature vectors to K new ones, capturing relations and preserving correspondence between each feature vector and each person.
- A relational scene representation. By stacking multiple relational layers, each with its own relationship graph, we build a scene representation encoding hierarchical

relationship representations. This representation is suitable for scenes of multiple related objects, such as in multi-person activity recognition.

- A novel autoencoder architecture that stacks multiple relational layers to jointly encode/decode each person's features based on relationship graphs. In unsupervised domains where no action labels are available, such representations can be used for scene retrieval based on nearest neighbour matching. A denoising autoencoder variant is also presented that infers missing people.
- Demonstrating the utility of these modules for (supervised) group activity recognition and (unsupervised) action/scene retrieval. We will publicly release our code[1]

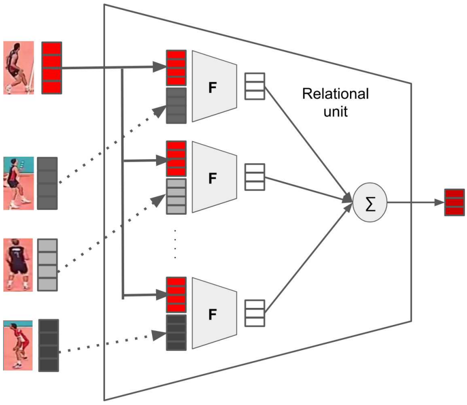


Fig. 2: Relatinal unit for processing one person inside a relational layer. The feature vector for a person (red) is combined with each of its neighbours'. Resultant vectors are summed to create a new feature vector for the person (dark red).

2 Related Work

We develop methods for multi-person activity recognition and retrieval by learning relational features. Below, we review related work in these areas.

Multi-person activity recognition: Recent deep learning approaches to multi-person activity recognition include Ibrahim et al. [9], which presents a 2-stage deep model. Person actions are modeled using a long short-term memory (LSTM) temporal layer. Scene dynamics are captured by adding a max-pooling layer which is fed to a higher-level LSTM. Ramanathan et al. [20] formulate an attention model to highlight key players in a scene, resulting in a weighted summation of person feature representations. Bagautdinov et al. [1] propose a joint model of action localization and group activity recognition. A multi-person object detection approach finds people and extracts their feature representations, which are linked based on Euclidean distance and fed to temporal recurrent

[1] https://github.com/mostafa-saad/hierarchical-relational-network

network. Shu et al. [25] extend this pipeline with an energy layer and confidence measure to consider reliability and numerical stability of the inference. Our work follows these 2-stage processing pipelines, but introduces a new relational layer that can learn compact relational representations for each person.

Image retrieval: Content-based retrieval for structured scenes is an active research area [23, 19, 28, 14]. Siddiquie et al. [26] extract multi-attributes and their correlations from a text query. Lan et al. [16] introduce queries that specify the objects that should be present in the scene, and their spatial relations (e.g., "car on the road"). Kim et al. [12] retrieve video clips that emphasize the progression of the text query. Johnson et al. [11] consider scene graph queries (objects and relationships). Xu et al. [29] generate scene graphs via a message passing neural network. In the realm of multi-person activity recognition, hard-coded representations of spatial relations have been developed previously [2, 15]. We show how our relational layers can be used in structured scene image retrieval, by matching frames of similar visual *structure* of people and their *actions*.

Relational networks: Recent work with deep networks includes capturing object relationships through aggregating with *every-pair-relation* models. Santoro et al. [24] introduce a relational network module that infers relationships between image objects. A multi-layer perceptron (MLP) learns the relationship of two objects, the scene is represented as summation of all object pairs. In a similar manner, Guttenberg et al. [8] use an MLP to learn a permutation-equivariant representation of a group of objects based on the relationship of every pair of objects. Inspired by these simple relation networks, we introduce our hierarchical relational network to build a compact relational scene representation, while preserving the correspondence between the feature representation and each person.

3 Proposed Approach

This paper introduces a Hierarchical Relational Network that builds a compact relational representation per person. Recent approaches [9, 20, 8] represent people in a scene then directly (max/average) pool all the representations into a single scene representation. This final representation has some drawbacks such as dropping relationships between people and destroying the individual person features. We tackle these challenges through a relational layer that jointly creates K person representations for the K people in a scene. By stacking multiple relational layers, we compactly encode hierarchical relationship representations. In the next subsections, we elaborate on the details of the Relational Network, then show its applications in supervised classification and unsupervised retrieval settings.

3.1 Hierarchical Relational Network

Our relational network for multi-person activity recognition processes a single video frame at a time. An input video frame has K initial person feature vectors (e.g., person detections with features extracted by a CNN) associated with multiple potential relationship graphs (e.g., based on spatial Euclidean distance thresholds). A single relational layer is fed with both K feature vectors and a relationship graph, and maps them to K new relational representations.

The building block for our model is a relational unit that processes an individual person in the scene. Each person's feature vector is mapped to a new representation by aggregating information from each neighbouring person in the relationship graph. This is accomplished via a network that processes the person combined with each neighbour, followed by aggregation. This relational unit is depicted Fig. 2.

Within one relational layer, every person in the scene is processed using this unit. This results in new feature representations for each person in the scene, capturing their individual features as well as those from his/her neighbours.

By stacking multiple layers, each with its own graph and relational unit parameters, we learn hierarchical relationship representations for the people. Pooling of the final person representations is used to construct the scene representation. An overview of our relational network for multi-person activity recognition in a single frame is shown in Fig. 3.

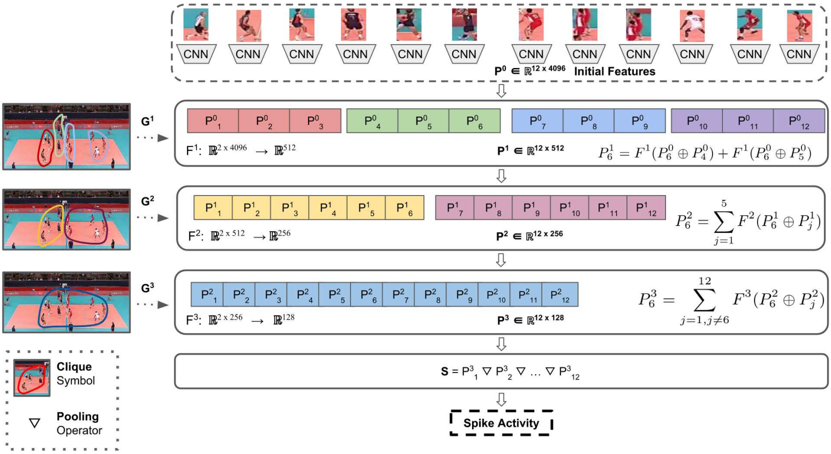

CNN
$P^0 \in \mathbb{R}^{12 \times 4096}$ Initial Features
G^1
P^0_1 P^0_2 P^0_3 P^0_4 P^0_5 P^0_6 P^0_7 P^0_8 P^0_9 P^0_{10} P^0_{11} P^0_{12}
F^1: $\mathbb{R}^{2 \times 4096} \rightarrow \mathbb{R}^{512}$
$P^1 \in \mathbb{R}^{12 \times 512}$
$P^1_6 = F^1(P^0_6 \oplus P^0_4) + F^1(P^0_6 \oplus P^0_5)$
G^2
P^1_1 P^1_2 P^1_3 P^1_4 P^1_5 P^1_6 P^1_7 P^1_8 P^1_9 P^1_{10} P^1_{11} P^1_{12}
F^2: $\mathbb{R}^{2 \times 512} \rightarrow \mathbb{R}^{256}$
$P^2 \in \mathbb{R}^{12 \times 256}$
$P^2_6 = \sum_{j=1}^{5} F^2(P^1_6 \oplus P^1_j)$
G^3
P^2_1 P^2_2 P^2_3 P^2_4 P^2_5 P^2_6 P^2_7 P^2_8 P^2_9 P^2_{10} P^2_{11} P^2_{12}
F^3: $\mathbb{R}^{2 \times 256} \rightarrow \mathbb{R}^{128}$
$P^3 \in \mathbb{R}^{12 \times 128}$
$P^3_6 = \sum_{j=1, j\neq 6}^{12} F^3(P^2_6 \oplus P^2_j)$
$S = P^3_1 \bigtriangledown P^3_2 \bigtriangledown \ldots \bigtriangledown P^3_{12}$
Spike Activity
Clique Symbol
▽ Pooling Operator

Fig. 3: Our relational network for group activity recognition for a single video frame. Given K people and their initial feature vectors, these vectors are fed to 3 stacked relational layers (of output sizes per person: 512, 256, 128). Each relational layer is associated with a graph G^ℓ (disjoint cliques in this example: layer 1 has 4 cliques, each of size 3; layer 3 is a complete graph). The shared MLP F^ℓ of each layer computes the representation of 2 neighbouring players. Pooling of the output K feature vectors is used for group activity classification.

Formally, given a video frame, the i^{th} person representation P_i^ℓ in the ℓ^{th} relational layer is computed as follows:

$$P_i^0 = CNN(I_i) \tag{1}$$

$$P_i^\ell = \sum_{j \in \mathcal{E}_i^\ell} F^\ell(P_i^{\ell-1} \oplus P_j^{\ell-1}; \theta^\ell) \tag{2}$$

where P_i^0 is the initial i^{th} person representation derived from a CNN on cropped image I_i, $\mathcal{E}_i^\ell$ is the set of relationship edges from the i^{th} person in the graph G^ℓ used for the ℓ^{th} layer, and $\oplus$ is the concatenation operator. $P_i^\ell \in \mathbb{R}^{N_\ell}$ where N_ℓ is the output size per-person for the ℓ^{th} layer.

The function F^ℓ is a shared MLP for the ℓ^{th} network layer with parameters θ^ℓ (end-to-end differentiable model). The MLP has input size $2N_{\ell-1}$ and output size N_ℓ. Given two concatenated vectors, F^ℓ maps them to a new vector capturing the given pair's content and relationship.

The relational layer feeds each edge in G^ℓ through its own shared MLP to compute the K new representations. Equation 2 computes a relationship representation between the i^{th} person and his/her neighbours. This network structure and the use of layer-wise shared parameters results in relationship representations per layer – treating each pair of people within one network layer equivalently. This results in efficient parameter reuse while letting the representation be driven by the graph structure at each layer. Importantly, this representation can also be used with any number of people K, including situations where K can vary per time step due to occlusions or false positive detections.

By stacking multiple compressive *relational layers*, each with its own graph, we can construct reduced dimension person features from one layer to another until a desired compact relational representation has been formed. The final scene representation S is the pooling of person representations from the last relational layer output and defined as:

$$S = P_1^L \triangledown P_2^L \triangledown \ldots \triangledown P_k^L \tag{3}$$

where P_i^L is the i^{th} person output representation of last relational layer L and $\triangledown$ is a pooling operator (such as vector concatenation or element-wise max pooling).

3.2 Supervised Learning: Group Activities

The activity of a group of people is a function of the persons' actions. We can utilize our model to represent each scene and learn its parameters in a supervised fashion. We utilize an Imagenet pre-trained VGG network [27] to represent each single person bounding box. The whole network is fine-tuned using action-labeled bounding boxes. Once trained, each person bounding box can be represented with the last layer in VGG19 (4096-d fc7 features).

Given the bounding boxes of the people in the scene in a video sequence, we recognize the overall multi-person activity. Each bounding box at the t^{th} frame is modeled and represented with an initial feature vector as explained above and fed to the relational network. The relational layer jointly maps the representations to ones that encode the relationship representation of a person based on connections to other people. To capture the temporal dynamics of the video scene, the output of the final *relational layer* is pooled to the t^{th} scene representation S_t and fed to an LSTM layer with a softmax output for group activity classification. Fig. 3 illustrates this model for a single frame.

3.3 Unsupervised Learning: Action Retrieval

Detailed annotation of individual person bounding boxes in video is a time-consuming process [7]. As an alternative, one could utilize unsupervised autoencoder mechanisms to learn feature representations for people in scenes. These representations could potentially be general-purpose: allowing comparison of person features based on relations and context for single-person action retrieval, and retrieval of scenes of similarly structured sets of actions.

Recent efforts in object recognition [18, 6] and temporal sequence learning [21, 17] aimed to learn effective feature representations in unsupervised encoding frameworks. In a similar vein, we propose unsupervised autoencoders that learn relational representations for all people in the scene.

Our relational layer is well-suited to this task since it: 1) encodes person relationships, 2) preserves action features for individual people, and 3) has compact size, efficient for retrieval. In other words, our scene representation is both efficient (compact size) and effective (relationship-based). Further, the model has the same parameter count as a simple autoencoder of a single person, as each layer has a shared network.

For the encoder, given K feature vectors for the people in the scene, we stack multiple relational layers of decreasing size that encode features to a final compact representation. The decoder is the inverse of these layers. That is, we again stack multiple relational layers of increasing size that decode a compressed feature vector to its original CNN representation. Each relational layer jointly maps a person representation from a given input size to a required output size considering graph connections. An Euclidean loss is computed between the initial K feature vectors and the corresponding decoded ones. An overview of the autoencoder model is shown in Fig. 4.

The reconstruction loss $\mathcal{L}$ of the input scene and its reconstructed one is given by:

$$\mathcal{L}(S_{cnn}, S'_{cnn}) = \sum_{i=1}^{K} \|P_i^0 - P_i^L\|^2 \tag{4}$$

where P_i^0 and P_i^ℓ are similar to Eq. 2 (but for a singel frame), S_{cnn} is the concatenation of the K initial feature vectors P_i^0, and S'_{cnn} is the reconstructed output of our network extracted from the last layer L. This novel autoencoder preserves features for individual people, so can be used for both scene and action retrieval.

Denoising Relational Autoencoder: What if some persons are missing in the scene (e.g., due to person detector failures, fast camera movement, or low image quality)? Denoising the input K feature vectors by dropping the *whole* vector for some of the persons allows our relational autoencoder to construct person representations from incomplete scenes. That is, our model infers the missing people from their context. To implement this, the input layer is followed by a dropout layer that drops a complete vector (not just subset of features) with probability P [22].

Retrieval: Given a single frame of K people, suppose we wish to search a video database for a matching frame with similar action structure. Note, the purpose is not retrieving a scene with the same overall activity, but a similar structured scene of actions. The pooled representation style, such as in [9], fits with group activity classification, but not with scene retrieval based on the matching of the actual actions due to losing person features for sake of a global scene representation. On the contrary, our representation for the scene preserves the individual person actions explicitly in a compact sized feature.

For the retrieval mechanism, we use a simple K-Nearest-Neighbour technique with a brute-force algorithm for comparison. To avoid comparison with each possible permutation, people are ordered based on the top corner (x, y) of a person's bounding box (on x first, and on y if tied). Euclidean distance is used to compare feature vectors.

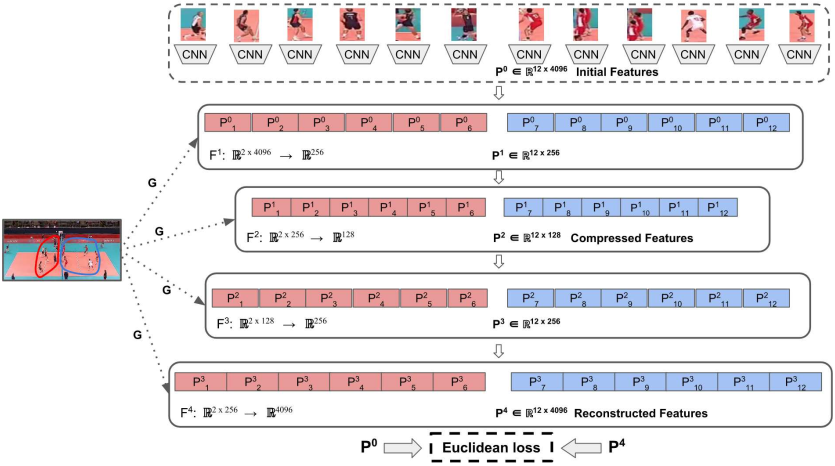

CNN CNN CNN CNN CNN CNN CNN CNN CNN CNN CNN CNN
$P^0 \in \mathbb{R}^{12 \times 4096}$ Initial Features
P^0_1 P^0_2 P^0_3 P^0_4 P^0_5 P^0_6 P^0_7 P^0_8 P^0_9 P^0_{10} P^0_{11} P^0_{12}
F^1: $\mathbb{R}^{2 \times 4096} \rightarrow \mathbb{R}^{256}$
$P^1 \in \mathbb{R}^{12 \times 256}$
G
P^1_1 P^1_2 P^1_3 P^1_4 P^1_5 P^1_6 P^1_7 P^1_8 P^1_9 P^1_{10} P^1_{11} P^1_{12}
F^2: $\mathbb{R}^{2 \times 256} \rightarrow \mathbb{R}^{128}$
$P^2 \in \mathbb{R}^{12 \times 128}$ Compressed Features
G
P^2_1 P^2_2 P^2_3 P^2_4 P^2_5 P^2_6 P^2_7 P^2_8 P^2_9 P^2_{10} P^2_{11} P^2_{12}
F^3: $\mathbb{R}^{2 \times 128} \rightarrow \mathbb{R}^{256}$
$P^3 \in \mathbb{R}^{12 \times 256}$
G
G
P^3_1 P^3_2 P^3_3 P^3_4 P^3_5 P^3_6 P^3_7 P^3_8 P^3_9 P^3_{10} P^3_{11} P^3_{12}
F^4: $\mathbb{R}^{2 \times 256} \rightarrow \mathbb{R}^{4096}$
$P^4 \in \mathbb{R}^{12 \times 4096}$ Reconstructed Features
P^0 Euclidean loss P^4

Fig. 4: Our relational autoencoder model. The relationship graph for this volleyball scene is 2 disjoint cliques, one for each team and fixed for all layers. K input person feature vectors, each of length 4096, are fed to a 4-layer relational autoencoder (sizes 256-128-256-4096) to learn a compact representation of size 128 per person.

4 Experiments

To demonstrate the power of our relational network, we evaluate it for two tasks: group activity recognition and action scene retrieval. The results are evaluated on the recent Volleyball Dataset [9]. The dataset consists of 4830 short clips gathered from 55 volleyball games, with 3493 training clips and 1337 for testing. Each clip is classified to one of 8 scene activity labels. Only the middle frame of each clip is fully annotated with the players' bounding boxes and their action labels (out of 9 actions). Clips of 10 frames (centered around the annotated middle frame) are used for the activity recognition task and the middle frame is used for the action scene retrieval task.

Our relational layer accepts free-form graph relationships. For volleyball, one suitable style is graphs of disjoint cliques based on person spatial locations. For example, in volleyball games there might be 3 potential graphs: **I)** All players are in 1 clique (1C), represents all pairwise relationships; **II)** each team can be a clique (2C); **III)** each team can be composed of 2 cliques, a total of 4 cliques (4C). We base our experiments on these clique-based groupings.

For the final scene pooling, instead of just max-pooling all persons, we use a slight variant [10] that reduces confusions between actions of the two team. Specifically, we max-pool each team individually, then concatenate the two representations. This is the default pooling strategy unless otherwise mentioned. In addition, due to the final person features' compact size, we could also do all-persons concatenation pooling. The concatenation pooling is neither effective nor efficient in other recent approaches [9] [25] due to the large dimensionality of the final person representation.

4.1 Group Activity Recognition

We refer to our activity recognition model as RCRG: *Relational Compact Representation for Group activity recognition*. RCRG is a 2-stage processing model and its input is clips of 10 timesteps, centered around the middle annotated frame. In the first stage, we fine-tune an ImageNet-pretrained VGG19 network using the annotated person bounding boxes (not a temporal model). This trained network is then used to represent each person bounding box using the penultimate network layer (fc7, 4096-d features). The person action recognition accuracy from the VGG19 model is 81%. In the second stage, K person representations are fed to our hierarchical relational network (associated with a relationship graph per layer) as in Fig. 3.

Table 1: Volleyball Dataset: Left table is for versions of our model using single frame (last row shows state-of-the-art using a single frame). Right table is for 10-timesteps input clips performance of our best models versus state-of-the-art.

Method	Accuracy
B1-NoRelations	85.1
RCRG-1R-1C	86.5
RCRG-1R-1C-!tuned	75.4
RCRG-2R-11C	86.1
RCRG-2R-21C	87.2
RCRG-3R-421C	86.4
RCRG-2R-11C-conc	**88.3**
RCRG-2R-21C-conc	86.7
RCRG-3R-421C-conc	87.3
Bagautdinov et al. [1]-single	83.8

Method	Accuracy
Bagautdinov et al. [1]	**90.6**
RCRG-2R-11C-conc	89.5
RCRG-2R-21C	89.4
Shu et al. [25]	83.3
Ibrahim et al. [10]	81.9

Baselines: We perform ablation studies with the following **non-temporal** (single frame) variants of our model to help us understand the performance of the model. The default pooling strategy is max-pooling unless **-conc** postfix is used to indicate concatenation pooling.

1. **B1-NoRelations:** In the first stage, the ImageNet-pretrained VGG19 network is fined tuned and a person is represented with fc7, 4096-d features. In the second stage, each person is connected to a shared dense layer of 128 features, then the person representations (each of length 128 features) are pooled, then fed to a softmax layer for group activity classification. This variant compresses person representations and represents the scene without inferring relationship representations.

2. **RCRG-1R-1C:** Same as previous variant, but the shared dense layer is replaced with a single relational layer (1R), all people in 1 clique (1C), i.e. all-pairs relationships. The layer maps each person from input size 4096 to 128 features jointly considering the given relationships.
3. **RCRG-1R-1C-!tuned:** Same as previous variant, but ImageNet-pretrained VGG19 without fine-tuning.
4. **RCRG-2R-11C:** Close to the RCRG-1R-1C variant, but uses 2 relational layers (2R) of sizes 256 and 128. The graphs of these 2 layers are 1 clique (11C) of all people. This variant and the next ones explore stacking layers with different graph structures.
5. **RCRG-2R-21C:** Same as the previous model, but the first layer has 2 cliques, one per team. The second layer is all-pairs relations (1C). **RCRG-2R-21C-conc** replaces the max pool strategy with concatenation pooling.
6. **RCRG-3R-421C:** Close to the previous model, but 3 relational layers (of sizes 512, 256 and 128) with clique sizes of the layers set to (4, 2, 1). The first layer has 4 cliques, with each team divided into 2 cliques. This model is in Fig. 3.

Implementation Details: We utilize the available dataset annotations for implementation. We follow Ibrahim et al. [9] to compute 10-frame tracklets of each person across the video sequence [3].

For training all the models and baselines, the same training protocols are followed using a Tesla K40C GPU (12 GB RAM) and Lasagne Framework [5]. Stochastic gradient descent is used train the model for 200 epochs and initial learning rate 10^{-4} with ADAM [13] optimizer, with fixed hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. We fine-tune the whole pre-trained VGG19 network [27] using batch-size 64 (small due to memory limits). For the relational model, a batch size of 250 is used. The input layer in our relational model is followed by a 50% dropout layer. Two-layer MLP networks are used of sizes N_ℓ. The first layer uses a linear activation function ($f(x) = x$) and the second uses ReLU non-linearities. Note, the models are end-to-end differentiable, but due to memory limits we implement it in a 2-stage style, similar to recent approaches.

In testing, only one shared person network is loaded and used by the K players to extract their features. The time complexity of a relational layer depends on the summation of the nodes degrees in the layer's graph. In other words, for each directed edge, the MLP of a layer is evaluated.

To determine graph cliques, we follow a simple approach [10]. People are ordered based on the upper left corner (x, y) of their bounding box (on x first, and on y if tied). Cliques are generated by sweeping this ordered list. For example, to divide 12 people to 4 cliques of equal size, each 3 consecutive people are grouped as a clique. More sophisticated grouping (e.g., color/motion clustering) or gating functions [4] would be potential extensions.

Results: Tables 1 compare the classification performance of our compact representation with the baselines and the state-of-the-art approaches.

Discussion: Our non-temporal models' performance is superior to state-of-the-art corresponding models and outperform compact baselines. Note even without temporal information this model is superior to 2 recent temporal models (In right Table 1). It seems from the results that stacking 2 layers is enough in this domain: in a volleyball scene inter-person relationships are strong. Max-pooling is effective at a scene level. Likely, this is due to the domain; a few players are the key actors, and max-pooling can keep the right features.

4.2 Experiments for Action and Scene Retrieval

We evaluate our retrieval model trained using unsupervised learning, termed RAER (*Relational AutoEncoder for Retrieval*). Our main model is shown in Fig. 4. It consists of 4 relational layers (256-128-256-4096 sizes) and it assumes the graph is 2 cliques (one per team) in all layers. We denote this structure by RAER-4L-2222C. This means, each team is compressed jointly, but all people per layer use the same shared relational MLP. Once the network is trained, each person is represented with 128 features from the compressed layer and used for scene and person retrieval.

Performance Measure: We consider two volleyball dataset frames as a correct match if the IoU (intersection over union) of the distributions of actions of the two frames is ≥ 0.5). For example, if the person actions of frame 1 are 7 people standing and 5 moving, and frame 2 are 4 standing, 6 moving, and 2 jumping then IoU = $\frac{4+5+0}{7+6+2} = 0.6$, hence a match.

Baselines: We compare with the following single-frame baseline models. One naive way to implement such a retrieval system is to learn a person action autoencoder, with its input and output a single person feature vector. Then concatenating the persons in the scene can be used for scene match. However, such direct reduction ignores all relationships in the scene ending with a weak scene representation. Another possibility is a direct concatenation of original persons feature vectors (e.g., 4096). Such a large scene representation may work in some domains, however, this large scene dimensionality is problematic.

1. **B1-Compact128:** Autoencoder with input/output of a single person feature vector of length 4096 from the fc7 layer of a pre-trained VGG19 network. The 4096-d vector is fed to network layers of sizes 256, 128, 256, 4096. The middle layer (128 features) is used as a compressed representation of the person. This network is structured similar to our model and of same compact person size (128 features) for fair comparison.
2. **B2-VGG19:** No autoencoder. Each single person is represented directly with a feature vector of length 4096 from the fc7 layer of a pretrained VGG19 network. Note that this baseline uses a much larger dimensionality (4096 vs. 128 features per person) and is especially problematic for representing scenes of many people.

Implementation Details: The same settings are used as Sec. 4.1 except the following. We trained these models *without* person action labels for 150 epochs and initial learning rate 10^{-4}. The MLP in the last relational layer ends with sigmoid non-linearities instead of ReLU. For person modeling, the ImageNet-pretrained VGG19 network is used as-is, without fine-tuning. The same setup is used for the Denoising Autoencoder, but with initial learning rate 10^{-3}.

Results: In this section we list our results for the retrieval tasks. We present the scene retrieval results, followed by single person retrieval. Then we discuss the performance of the models.

Table 2 compares the scene retrieval performance of our relational autoencoder with the baselines. We compute the Hit@K measure for $K \in \{1, 2, \ldots, 5\}$. Specifically, given a query frame, the frame is encoded using the autoencoder model and the closest K matches in the database are retrieved. Recall, two frames are a match if the IoU of their actions $\geq$ threshold (0.5). Mean average precision is also reported: mean of the average precision values for each image query where Euclidean distance is used as the confidence indicator. The training and testing sets are the ground truth annotated scenes in the Volleyball Dataset. Results indicate how this novel architecture is capable of capturing the context and encoding it within each person. Surprisingly, our model even beats the uncompressed VGG19, though VGG should be much stronger due to its size and sparsity.

Table 2: Scene retrieval compared to baselines.

Method	Hit@1	Hit@2	Hit@3	Hit@4	Hit@5	mAP
B1-Compact128	49.4	68.7	80.4	87.7	91.4	35.4
B2-VGG19	55.0	73.9	82.7	87.5	91.5	36.4
RAER-4L-2222C	**57.4**	**76.7**	**85.3**	**90.4**	**93.3**	**36.8**

In Table 3, we explore variants of our scene retrieval model. Specifically, we try 2 models with only 2 relational layers (128, 4096): One of these models uses 1 clique in all layers (RAER-2L-11C, all pair relationships) and the second uses 2 cliques (RAER-2L-22C, all pairs within a team). The complex version (RAER-4L-4224C) is 2 layers as our main model, but layer cliques are (4, 2, 2, 4). This means the decoder has to learn how to decode such hierarchical information.

Table 3: Scene retrieval compared to model variants.

Method	Hit@1	Hit@2	Hit@3	Hit@4	Hit@5	mAP
RAER-2L-11C	56.8	74.9	84.5	89.8	92.6	**36.8**
RAER-2L-22C	56.9	75.6	84.9	90.0	**93.3**	36.7
RAER-4L-4224C	55.8	76.1	84.0	88.9	92.7	36.6
RAER-4L-2222C	**57.4**	**76.7**	**85.3**	**90.4**	**93.3**	**36.8**

In Table 4, we show the results for the Denoising Autoencoder when a person might be missing with probability 0.5 in the test data.

Table 5 compares the person retrieval performance of using the same relational autoencoder model with the baselines. The training and testing sets are the ground truth bounding boxes of annotated actions in the Volleyball Dataset. Note that the Volleyball dataset consists of 9 action labels, with standing class representing $\approx$70% of the action

Table 4: Scene Retrieval using Denoising Autoencoder (-D) with 50% possible drop for people in test data for models and baselines. Our model is robust; the No Autoencoder model performance drops significantly.

Method	Hit@1	Hit@2	Hit@3	Hit@4	Hit@5	mAP
B1-Compact128-D	38.1	58.8	70.5	78.2	84.7	34.6
B2-VGG19-D	34.0	51.1	62.2	70.0	76.0	34.9
RAER-4L-2222C-D	**43.0**	**65.0**	**78.7**	**85.8**	**90.7**	**35.2**

labels, so a retrieval system that keeps retrieving standing samples will score high results. To avoid that, the standing class is removed from both the training and test sets in the person retrieval task. After training the model, we extract the compressed person representations for each person action and build a retrieval model for them. Results indicate that our compact person representation works well and beats the alternative compression baseline.

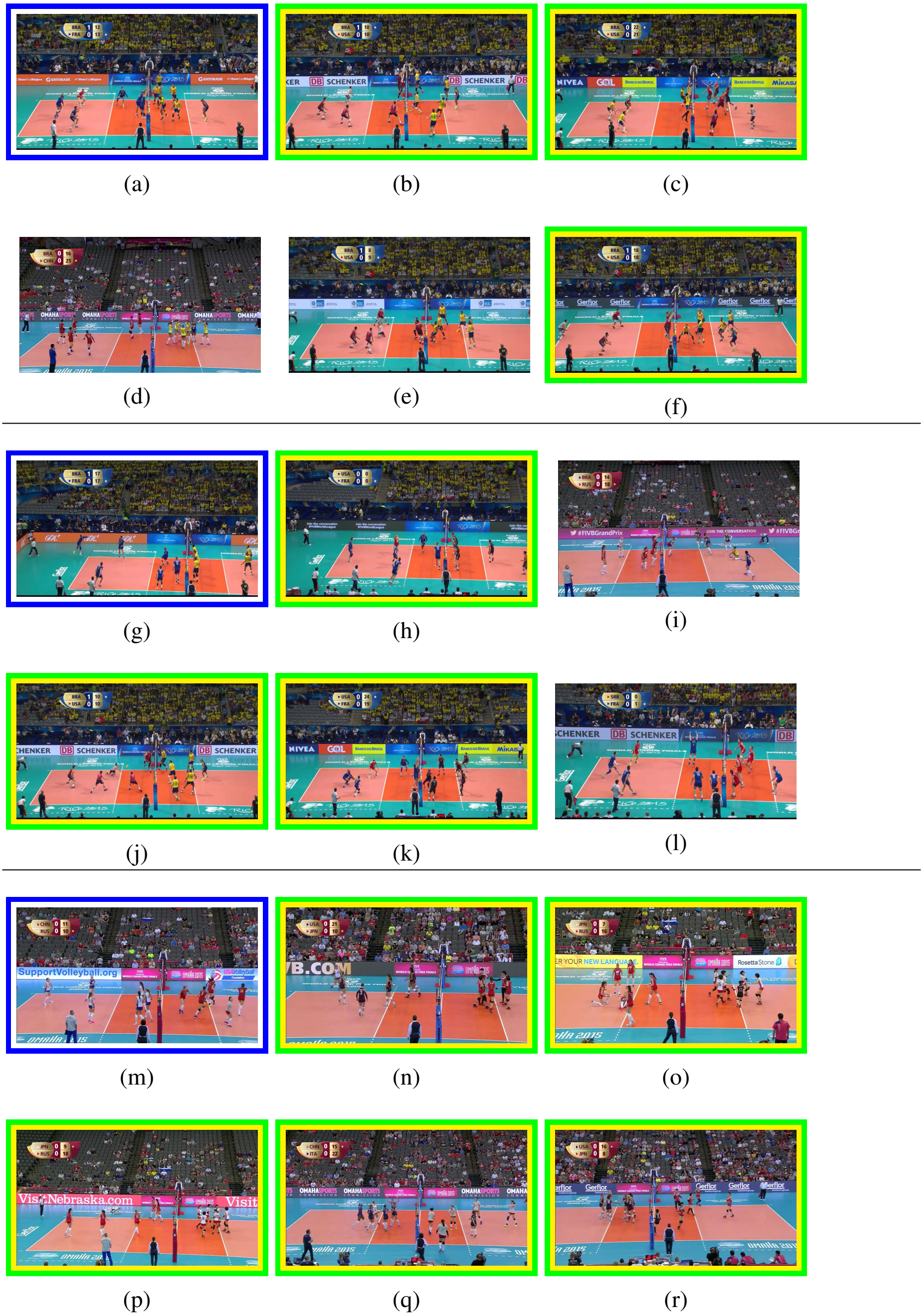
Table 5: Person Retrieval on Volleyball Dataset: Hit@K results of our method and baselines. Last column is mean average precision of query results. Our model outperforms the normal autoencoder model, and is competitive with a 32x larger sparse representation.

Method	Hit@1	Hit@2	Hit@3	Hit@4	Hit@5	mAP
B1-Compact128-P	37.7	54.7	64.6	71.7	76.4	22.8
B2-VGG19-P	**47.3**	**63.2**	**72.1**	**77.4**	**81.2**	25.4
RAER-2L-11C-P	45.5	62.2	70.9	76.1	80.1	**25.8**
RAER-4L-2222C-P	42.6	58.3	68.3	73.7	77.8	25.2

Discussion: The high Hit@K results indicate that the autoencoder approach works well for this task. From the scene and action retrieval results, we notice that our relational autoencoder outperforms the normal autoencoder model of the same structure and compression size due to encoding/decoding of person relationships. Of particular note, the autoencdoer outperforms high-dimensional VGG features for scene retrieval. We hypothesize that this is due to the ability of the relational layers to capture contextual information among people in the scene. Fig 5 visualizes scene retrieval results.

5 Conclusion

We proposed a hierarchical relational network for learning feature representations. The network can be used in both supervised and unsupervised learning paradigms. We utilized this network for group activity recognition, based on the final compact scene layer. We also showed how the relational layer can be the main building block in novel autoencoder models that jointly encode/decode each person's feature representation using a shared memory. Results in both tasks demonstrate the effectiveness of the relational network. The relationship graph associated with each layer allows explicit relationship consideration that can be applied to other visual understanding tasks.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

Fig. 5: Visualizations of scene retrieval using our relational autoencoder. Each 2 rows are a query: Query image first (blue box), followed by the closest 5 retrievals. Green Framed boxes are correct matches. The last query is for *Right team winpoint event*, and its results are 3 consecutive *Right team winpoint events* followed by 2 *Left team winpoint events*.

References

1. Bagautdinov, T.M., Alahi, A., Fleuret, F., Fua, P., Savarese, S.: Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
2. Choi, W., Shahid, K., Savarese, S.: Learning context for collective activity recognition. In: Computer Vision and Pattern Recognition (CVPR) (2011)
3. Danelljan, M., Hger, G., Shahbaz Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference (BMVC) (2014)
4. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
5. Dieleman, S., Schlter, J., Raffel, C., Olson, E., Snderby, S.K., Nouri, D., et al.: Lasagne: First release. (Aug 2015). https://doi.org/10.5281/zenodo.27878, http://dx.doi.org/10.5281/zenodo.27878
6. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: International Conference on Computer Vision (ICCV) (2015)
7. Gu, C., Sun, C., Ross, D.A., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: arXiv (2017)
8. Guttenberg, N., Virgo, N., Witkowski, O., Aoki, H., Kanai, R.: Permutation-equivariant neural networks applied to dynamics prediction. arXiv preprint arXiv:1612.04530 (2016)
9. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: Hierarchical deep temporal models for group activity recognition. arXiv preprint arXiv:1607.02643 (2016)
11. Johnson, J., Krishna, R., Stark, M., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Image retrieval using scene graphs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
12. Kim, G., Moon, S., Sigal, L.: Ranking and retrieval of image sequences from multiple paragraph queries. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2014)
14. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalanditis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision (IJCV) **123**, 32–73 (2017)
15. Lan, T., Wang, Y., Mori, G., Robinovitch, S.N.: Retrieving actions in group contexts. In: European Conference on Computer Vision (ECCV) Workshops (2010)
16. Lan, T., Yang, W., Wang, Y., Mori, G.: Image retrieval with structured object queries using latent ranking SVM. In: European Conference on Computer Vision (ECCV) (2012)
17. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: International Conference on Computer Vision (ICCV) (2017)
18. Pathak, D., Krhenbhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Computer Vision and Pattern Recognition (CVPR) (2016)
19. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

20. Ramanathan, V., Huang, J., Abu-El-Haija, S., Gorban, A., Murphy, K., Fei-Fei, L.: Detecting events and key actors in multi-person videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
21. Ramanathan, V., Tang, K., Mori, G., Fei-Fei, L.: Learning temporal embeddings for complex video analysis. In: International Conference on Computer Vision (ICCV) (2015)
22. Ravanbakhsh, S., Schneider, J.G., Póczos, B.: Deep learning with sets and point clouds. In: International Conference on Learning Representations (ICLR) - workshop track (2017)
23. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
24. Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.P.: A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427 (2017)
25. Shu, T., Todorovic, S., Zhu, S.: CERN: confidence-energy recurrent network for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
26. Siddiquie, B., Feris, R.S., Davis, L.S.: Image ranking and retrieval based on multi-attribute queries. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2014)
28. Stewénius, H., Gunderson, S.H., Pilet, J.: Size matters: Exhaustive geometric verification for image retrieval. In: European Conference on Computer Vision (ECCV) (2012)
29. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: CVPR (2017)