Stable Bidirectional Graph Convolutional Networks for Label-Frugal Skeleton-based Recognition

Hichem Sahbi
Sorbonne University, CNRS, LIP6
F-75005, Paris, France
hichem.sahbi@sorbonne-universite.fr

Abstract

Skeleton-based action recognition is a major challenge in computer vision. In particular, solutions based on graph convolutional networks (GCNs) have demonstrated notable performance, but their success is reliant on the availability of large collections of hand-labeled skeleton sequences. However, in real-world applications, these sequences are often scarce, prompting the exploration of label-frugal GCN models. In this paper, we introduce a novel label-efficient GCN model for skeleton-based action recognition. As a first contribution, we devise a new acquisition function that allows us to design exemplars (a few candidate data for labeling) using an adversarial objective function that mixes representativity, diversity and uncertainty of these exemplars. As a second contribution, we make our designed GCNs bidirectional and stable, allowing them to map data from ambient to latent spaces (and vice-versa) where the inherent distribution of the learned exemplars is more easily captured. Extensive experiments conducted on two challenging skeleton-recognition datasets, show a substantial gain of our frugally designed GCNs against the related work.

1. Introduction

Skeleton-based recognition consists in analyzing articulated body scenes by extracting joint locations and modeling their spatio-temporal interactions. Early methods rely on handcrafted features [14, 17, 26, 34, 35, 38, 58, 59], such as joint angles and relative distances, fed as inputs to classifiers including support vector machines and hidden Markov models [11, 47, 49, 50, 57], or combined with manifold learning techniques [15, 16, 21, 62]. With the resurgence of deep learning [9, 18, 19, 23, 33], recurrent neural networks, notably LSTMs and GRUs [7, 27, 28, 30, 61, 63], gained prominence for capturing the temporal dynamics in skeletal sequences. Subsequently, Graph Convolutional Networks (GCNs) emerged, leveraging the inherent graph structure of skeletons to learn spatial relationships between joints [32, 39, 40]. Attention-based models [25, 29, 37, 41, 46], incorporating GCNs, have also demonstrated significant performance improvements by effectively modeling long-range dependencies and capturing complex motion patterns.

The efficacy of learning-based methods in skeleton-based recognition is fundamentally dependent on the availability of extensive, diverse datasets carefully hand-labeled with skeleton sequences. However, the acquisition of such large-scale datasets presents a significant challenge, requiring substantial time and labor. Several strategies have been proposed to mitigate data and label scarcity, including augmentation techniques [45] that artificially expand data size and variability. Furthermore, few-shot and transfer learning approaches [3] leverage pre-existing knowledge from related domains, while self-supervised learning methods [48] seek to extract inherent patterns from unlabeled data. Despite their contributions, the relative success of these solutions often relies on the implicit assumption that the existing knowledge, whether derived from augmented data or pre-trained models, is sufficient to bridge the *accuracy gap*. In reality, the quality and relevance of labeled data remain paramount.

In contrast to the foregoing passive learning approaches, active learning [43] presents a more efficient and targeted strategy for dataset construction. Active learning effectively selects the most informative samples for labeling, thereby maximizing the model's learning potential with minimal human annotation effort. By iteratively querying an oracle (human annotator) for labels on the most uncertain or representative samples, active learning prioritizes the acquisition of data that may yield the greatest improvement in model accuracy. This approach not only reduces the overall labeling burden but also ensures that the labeled dataset is optimally tailored to the specific recognition task. In scenarios where data or label acquisition is costly or time-consuming, active learning offers a compelling alternative, as it directly addresses the critical need for high-quality, relevant labeled data.

The selection of informative data within active learning revolves around identifying samples that maximize a model's learning potential. Strategies such as query-by-committee [44], expected model change maximization [4] and deep reinforcement learning [8] have emerged as powerful techniques and have been explored to enhance the informativeness of selected samples. These strategies typically integrate measures of uncertainty [2, 6, 10, 13, 48, 56] and diversity [1, 52] in different contexts [5, 24, 31, 36, 51]. Uncertainty-based strategies, such as margin sampling and entropy-based criteria [20, 53], prioritize samples where the model exhibits low confidence in its predictions, thus highlighting areas where further training is most beneficial. Diversity-based methods, including coverage maximization [22, 55] and core-set selection [42], aim to select samples that span the breadth of the data distribution, ensuring the model is exposed to a wide range of data variations. Complementary to these, representativeness-based approaches [54] seek samples that closely mirror the overall data distribution, fostering a balanced learning process. While these criteria offer valuable insights, many current solutions rely on heuristic-driven approaches, which may lack theoretical rigor. A more robust approach would involve developing selection criteria grounded in probabilistic frameworks, enabling the identification of truly optimal subsets. Such frameworks would not only enhance the efficiency of active learning but also provide a more principled way for constructing highly informative datasets.

Considering the aforementioned issues, we introduce in this paper a label-efficient GCN for skeleton-based recognition. The contribution of the proposed method resides in a novel principled probabilistic framework that designs unlabeled exemplars (candidate samples for labeling) instead of sampling them from a fixed pool of unlabeled data. These exemplars are obtained as an interpretable solution of an objective function mixing data representativity, diversity and uncertainty. Our proposed framework designs these exemplars using a novel stable and invertible bidirectional GCN that allows mapping input graphs (lying on highly nonlinear manifolds) from ambient (input) to latent spaces where learning these exemplars becomes more tractable; indeed, with the proposed GCNs, data in the latent space follow a standard probability distribution (namely gaussian) whose sampling and search is more tractable compared to the arbitrary distributions in the ambient space. Once learned, these exemplars are mapped back to the input space thanks to the invertibility and stability of our GCNs. In sum, the proposed framework allows designing bidirectional GCNs exhibiting both strong classification and exemplar design capabilities — including at frugal data regimes — without requiring auxiliary generative networks. Extensive experiments, conducted on two challenging skeleton-based recognition tasks, show the outperformance of our label-efficient method compared to the related work.

2. Display Model

Our proposed Active Learning (AL) framework comprises two core components: *display* and *learning* models. The display model defines an acquisition function to identify the most informative unlabeled data points, which are then presented to an oracle for labeling. Then, the learning model retrains a label-efficient classifier using the newly acquired labels. These two steps are iteratively executed until a predetermined classification accuracy is achieved or a labeling budget is exhausted. Formally, let $\mathcal{U} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^p$ be the pool of unlabeled data. At each AL iteration $t \in \{0, \dots, T-1\}$, the *display* model, as detailed in section 2.1, builds a subset $\mathcal{D}_t$—termed the display set—which is used to query the oracle for corresponding labels $\mathcal{Y}_t$. A classifier f_t is then trained on the cumulative labeled dataset $\bigcup_{k=0}^{t}(\mathcal{D}_k, \mathcal{Y}_k)$. Our first contribution (introduced in section 2.1) is based on a novel model that builds in a *flexible way* displays instead of sampling fixed ones from $\mathcal{U}$.

2.1. Display model design

Our proposed method is adversarial and consists in selecting the most diverse, representative, and uncertain data to effectively *challenge* the current classifier f_t, leading to an improved classifier f_{t+1} in the subsequent AL iteration. Instead of directly sampling the display set $\mathcal{D}_{t+1}$ from the unlabeled pool $\mathcal{U}$, we use a probabilistic framework to construct $\mathcal{D}_{t+1}$ (denoted for short as $\mathcal{D}$). Let $\mathbf{X} \in \mathbb{R}^{p\times n}$ and $\mathbf{D} \in \mathbb{R}^{p\times K}$ be two matrices representing $\mathcal{U}$ and $\mathcal{D}$, respectively, where $K = |\mathcal{D}|$. In order to construct the display $\mathbf{D}$, we assign a conditional probability distribution to each column $\mathbf{D}_k$, quantifying the membership (or contribution) μ_{ik} of each $\mathbf{x}_i \in \mathcal{U}$ in forming $\mathbf{D}_k$. The memberships $\mu = \{\mu_{ik}\}_{ik}$ and the display $\mathbf{D}$ are found by minimizing the following constrained objective function

$$\min_{\mu\in\Omega,\mathbf{D}} \mathbf{tr}(\mu\, d(\mathbf{X},\mathbf{D})^\top) + \alpha \sum_{k,k'}^{K,N} \exp\left\{-\frac{1}{\sigma}\|\mathbf{D}_k - \mathbf{H}_{k'}\|_2^2\right\} + \beta\, \mathbf{tr}(\mathbf{D}^\top\mathbf{D}) + \gamma\, \mathbf{tr}(\mu^\top \log\mu), \tag{1}$$

being $\Omega = \{\mu : \mu \geq 0, \mathbf{1}_n^\top \mu = \mathbf{1}_K\}$ a convex set constraining μ to be column-stochastic (i.e., each column represents a conditional probability distribution), where $\mathbf{1}_K$ and $\mathbf{1}_n$ are vectors of K and n ones, respectively, and $^\top$ denotes the transpose. The objective function (in Eq. 1) comprises four terms

- **Representativity:** this term minimizes the discrepancy between the designed exemplars in $\mathbf{D}$ and the data distribution in $\mathcal{U}$. This ensures that the oracle's annotations are based on realistic exemplars, preventing the selection of trivial or meaningless data.

- **Diversity:** it maximizes the dissimilarity between the N previously selected exemplars (represented by a matrix $\mathbf{H}$) and the K currently selected exemplars (matrix $\mathbf{D}$). This enforces that new exemplars are *as distinct as possible* from the previous ones.
- **Uncertainty:** it measures the ambiguity associated with exemplars in $\mathbf{D}$. It encourages the selection of exemplars near the decision boundaries of the learned classifiers. This term also acts as a regularizer on $\mathbf{D}$. Minimizing this term identifies ambiguous exemplars, which are crucial for reducing model uncertainty, and accelerating convergence to well-defined decision functions.
- **Regularization of μ:** this term regularizes μ, promoting flat conditional probabilities $\mu = \{\mu_{ik}\}_{ik}$ in the absence of a priori knowledge about the other three terms.

All these terms are weighted by $\alpha, \beta, \gamma \geq 0$, whose setting is discussed later.

Proposition 1. *The optimality conditions of Eq. 1 lead to the solution as the fixed-point of*

$$\begin{array}{rcl} \mu^{(\tau+1)} & := & \hat{\mu}^{(\tau+1)}\ \mathbf{diag}\big(\mathbf{1}_n^\top \hat{\mu}^{(\tau+1)}\big)^{-1} \\ \mathbf{D}^{(\tau+1)} & := & \hat{\mathbf{D}}^{(\tau+1)}\ \big(\mathbf{diag}(\mathbf{1}_n^\top \mu^{(\tau)}) + \beta \mathbf{I}\big)^{-1}, \end{array} \tag{2}$$

being $\hat{\mu}^{(\tau+1)}$, $\hat{\mathbf{D}}^{(\tau+1)}$ respectively

$$\begin{array}{l} \exp\big\{-\frac{1}{\gamma} d(\mathbf{X}, \mathbf{D}^{(\tau)})\big\}, \\ \\ \mathbf{X}\ \mu^{(\tau)} - \dfrac{2\alpha}{\sigma}\big(\mathbf{D}^{(\tau)}\ \mathbf{diag}(\mathbf{1}_N' \mathbf{S}) - \mathbf{H}\mathbf{S}\big), \end{array} \tag{3}$$

where $\mathbf{S}$ equates (with $\mathbf{D}^{(\tau)}$ written for short as $\mathbf{D}$)

$$\exp\left\{-\frac{1}{\sigma}\big(\mathbf{1}_N \mathbf{diag}(\mathbf{D}^\top \mathbf{D})^\top + \mathbf{diag}(\mathbf{H}^\top \mathbf{H})\mathbf{1}_K^\top - 2\mathbf{H}^\top \mathbf{D}\big)\right\}, \tag{4}$$

here $\mathbf{S}$ is a similarity matrix between $\mathbf{D}$ and $\mathbf{H}$, $\mathbf{1}_N$ is a vector of N ones, and **diag** *maps a vector to a diagonal matrix.*

Due to space limitation, the detailed proof, derived from the optimality conditions of Eq. 1's gradient, is omitted. More notably, the solution for μ in Eq. 3 demonstrates an inverse relationship between data distances and membership values: low distances result in high memberships of input data $\mathbf{X}$ to exemplars $\mathbf{D}$, and vice versa. The solution for $\mathbf{D}$ shows that each exemplar $\mathbf{D}_k$ is a combination of two terms. The first one is a normalized linear combination of data weighted by their memberships to $\mathbf{D}_k$. The second term, controlled by α, disrupts $\mathbf{D}_k$ to maximize its dissimilarity from previously selected exemplars $\mathbf{H}$.

Initially, $\mu^{(0)}$ and $\mathbf{D}^{(0)}$ are set to random values. In practice, the iterative procedure converges to an optimal solution $(\tilde{\mu}, \tilde{\mathbf{D}})$ within a few iterations. This solution defines the subsequent display $\mathcal{D}_{t+1}$ used to train f_{t+1} (see algorithm. 1). The parameters α and β are set to balance the impact of their respective terms, specifically $\alpha = \frac{1}{KN}$ and $\beta = \frac{1}{Kp}$. In Eq. 3, σ is set proportional to α in order to absorb the former by the latter, and to reduce the total number of hyperparameters. The hyperparameter γ, which scales the exponential function in $\hat{\mu}^{(\tau+1)}$, is iteration-dependent and proportional to the input of that exponential, namely $\log(\hat{\mu}^{(\tau+1)})$; therefore, $\gamma = \frac{1}{nK} \| \log(\hat{\mu}^{(\tau+1)}) \|_1$ in practice.

Now considering the aforementioned AL formulation, two variants of the proposed solution are considered in this paper. The first one finds exemplars using the above formulation directly in the ambient (input) space, while the second one finds the exemplars in the latent space, and maps them back to the ambient space thanks to the invertibility and also stability of the learned GCNs (as shown in section 3). As shown subsequently, relying on invertible and stable GCN mapping leads to an extra gain in AL performances as also shown later through experiments.

Algorithm 1: Exemplar learning

Input: $\mathcal{U}$, $\mathcal{D}_0 \subset \mathcal{U}$, budget T.
Output: $\cup_{t=0}^{T-1}(\mathcal{D}_t, \mathcal{Y}_t)$ and $\{f_t\}_t$.

for $t := 0$ **to** $T-1$ **do**
- $\mathcal{Y}_t \leftarrow \text{oracle}(\mathcal{D}_t)$;
- $f_t \leftarrow \arg\min_f \text{Loss}(f, \cup_{k=0}^{t}(\mathcal{D}_k, \mathcal{Y}_k))$ (loss in Eqs. 6 or 7);
- $\tau \leftarrow 0$; $\hat{\mu}^{(0)} \leftarrow$ random; $\hat{\mathbf{D}}^{(0)} \leftarrow$ random;
- Set $\mu^{(0)}$ and $\mathbf{D}^{(0)}$ using Eqs. (2) and (3);
- **while** $(\|\mu^{(\tau+1)} - \mu^{(\tau)}\|_1 + \|\mathbf{D}^{(\tau+1)} - \mathbf{D}^{(\tau)}\|_1 \geq \epsilon \wedge \tau < \textit{maxiter})$ **do**
  - Set $\mu^{(\tau+1)}$ and $\mathbf{D}^{(\tau+1)}$ using Eqs. (2) and (3);
  - $\tau \leftarrow \tau + 1$;
- $\tilde{\mu} \leftarrow \mu^{(\tau)}$; $\tilde{\mathbf{D}} \leftarrow \mathbf{D}^{(\tau)}$.

3. Learning Model

As previously discussed, the efficacy of the active learning process hinges on the suitability of the display model. Specifically, the generated displays should accurately reflect the data distribution within the input space. However, when dealing with complex, nonlinear data distributions, the display model defined in Eq. 1 may encounter a significant limitation. Data lying on nonlinear manifolds pose a challenge for ensuring that the generated displays remain consistent with these manifolds. Consequently, in the following section, we revisit GCNs and introduce — as a second contribution —– a novel learning model designed to address this limitation by making our trained GCNs bidirectional, invertible and stable (see Fig. 1).

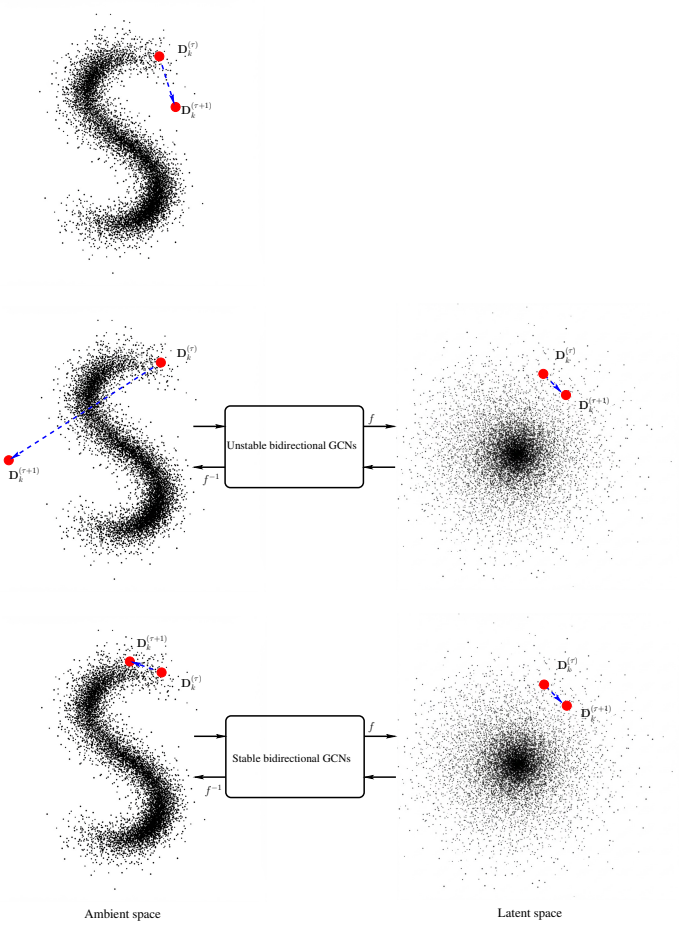

Figure 1. This figure shows the display model under three different configurations: (top row) when exemplars are learned in the input (ambient) space, fixed-point iteration (FPI) may lead to out-of-distribution (OOD) exemplars, (middle row) when using the latent space, any slight update of the exemplars (using FPI), may lead to OOD data when the learned bidirectional GCNs are not stable, (bottom row) in contrast, when using stable bidirectional GCNs, slight updates in the latent space also result in slight updates in the ambient space preventing OODs. "Red-colored disks" stand for exemplars while "blue-colored arrows" stand for the direction of updates obtained with FPI (better to zoom the PDF version).

3.1. Graph convnets at a glance

Let $\{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_i$ denote a collection of graphs, where $\mathcal{V}_i$ and $\mathcal{E}_i$ represent the node and edge sets of $\mathcal{G}_i$, respectively. For simplicity, consider a single graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from this collection; $\mathcal{G}$ is associated with a signal $\{\psi(v) \in \mathbb{R}^s\}$ for all nodes $v \in \mathcal{V}$, and an adjacency matrix $\mathbf{A}$. GCNs aim to learn a set of C filters, represented by the matrix $\mathbf{W} \in \mathbb{R}^{s \times C}$, that define a convolution operation on the m nodes of $\mathcal{G}$, where $m = |\mathcal{V}|$. This operation is defined as $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}} = g\big(\mathbf{A}\ \mathbf{U}^\top\ \mathbf{W}\big)$, where $\mathbf{U} \in \mathbb{R}^{s \times m}$ is the graph signal, and $g(\cdot)$ is a nonlinear activation function applied entrywise. In this operation, the input signal $\mathbf{U}$ is projected using the adjacency matrix $\mathbf{A}$, effectively aggregating signals from the neighbors of each node v. The entries of $\mathbf{A}$ can be either pre-defined or learned. Thus, $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$ can be interpreted as a two-layer convolutional block. The first layer aggregates signals from the neighborhood $\mathcal{N}(\mathcal{V})$ of each node v by multiplying $\mathbf{U}$ with $\mathbf{A}$, whilst the second layer performs the convolution by multiplying the resulting aggregated signals with the C filters in $\mathbf{W}$.

3.2. Proposed stable bidirectional GCNs

We formally subsume a given GCN as a multi-layered neural network f with weights $\theta = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$, where L represents the depth of the network. The weight tensor for the ℓ-th layer is denoted as $\mathbf{W}_\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$, being d_ℓ its ℓ-th layer dimension. The output of a given layer, denoted as ϕ_ℓ, is defined as $\phi^\ell = g_\ell({\mathbf{W}_\ell}^\top\ \phi^{\ell-1})$, $\ell \in \{2, \ldots, L\}$, where g_ℓ is a nonlinear activation function. Without loss of generality, we omit the bias in the definition of ϕ^ℓ.

In this section, we are interested in designing invertible and stable bidirectional networks. The invertibility (bijection) of the function $f : \mathbb{R}^p \to \mathbb{R}^q$ ensures the existence of a *one-to-one* correspondence between $\mathbb{R}^p$ and $\mathbb{R}^q$ (with necessarily $p = q$)[1] guaranteeing that (i) no two distinct network inputs, ϕ_1^1 and ϕ_1^2, map to the same output ϕ_L, and (ii) for every output ϕ_L, there exists at least one input ϕ_1 such that $f(\phi_1) = \phi_L$; effectively, making the trained GCNs bidirectional. Stability pushes invertibility "one step further" to guarantee that f^{-1} — when evaluated on a given targeted latent distribution (e.g., gaussian) — does not diverge from the ambient (input) distribution, and vice versa.

Definition 1 (Stability)**.** *A bidirectional network $f : \mathbb{R}^p \to \mathbb{R}^q$ is called bi-Lipschitzian (or KM-Lipschitzian), if f is K-Lipschitzian and its inverse f^{-1} is M-Lipschitzian. The KM-Lipschitz constant of a bidirectional network is defined as $K \times M$.*

In general, making both K and M small for any given nonlinear function is challenging [12]; thereby making the KM constant small is also challenging. However, considering our following bidirectional network design, it becomes possible under specific conditions to make KM small (namely close to 1 as a result of our subsequent proposition).

Proposition 2. *Provided that (i) the entrywise activations $\{g_\ell(.)\}_{\ell=2}^L$ are bijective in $\mathbb{R}^p$, (ii) $l \leq |g'_\ell(.)| \leq u$, and (iii) the condition numbers of the weight matrices in θ are bounded by κ, then the bidirectional network f is KM-Lipschitzian with*

$$KM = (\kappa\, u/l)^{L-1}. \tag{5}$$

[1] As the output of f depends on the number of classes, a simple trick consists in adding fictional outputs to match any targeted dimensionality (similarly for other layers).

Sketch of the proof is given in the appendix. More importantly, following the aforementioned proposition, when f is invertible in $\mathbb{R}^p$, then one may derive $f^{-1}(\phi^L) = \phi^1$ being $\phi^{\ell-1} = (\mathbf{W}_\ell^\top)^{-1} g_\ell^{-1}(\phi^\ell)$. The condition number of a matrix $\mathbf{W}_\ell$ — defined as $\|\mathbf{W}_\ell\|_2.\|\mathbf{W}_\ell^{-1}\|_2$ — measures how sensitive $\mathbf{W}_\ell$ to small changes in $\phi^{\ell-1}$ and ϕ^ℓ. A small condition number indicates a well-conditioned matrix $\mathbf{W}_\ell$. When κ, l and u are close to 1, then $KM \approx 1$ meaning that the bidirectional network f is 1-Lipschitzian so any slight update of exemplars in the latent space (with the fixed-point iteration in Eq. 2) will also result into a slight update of these exemplars in the ambient space when applying f^{-1}. This eventually leads to stable exemplar design in the ambient space, i.e., they follow the actual distribution of data manifold.

As the Lipschitz constant of f is $\prod_\ell \|\mathbf{W}_\ell\|_2.\left|g'_\ell\right|$, and for f^{-1} is $\prod_\ell \|(\mathbf{W}_\ell^\top)^{-1}\|_2 \left|g_\ell^{-1'}\right|$ (see proof in appendix), the sufficient conditions that guarantee that the bidirectional network is KM-Lipschitzian (with small KM) corresponds again to (1) small condition numbers $\{\|\mathbf{W}_\ell\|_2 \times \|\mathbf{W}_\ell^{-1}\|_2\}_\ell$, and (2) $l, u \approx 1$ (with $l < u$ in order to guarantee the nonlinearity of f). Hence, by design, conditions (1)+(2) could be satisfied by choosing the slope of the activation functions to be close to one (in practice $u = 0.99$ and $l = 0.95$ corresponding respectively to the positive and negative slopes of the leaky-ReLU[2]), and also by constraining all the weight matrices to have a low condition number. This is obtained by adding a regularization term, to the cross-entropy (CE) loss, when training GCNs, as

$$\min_{\{\mathbf{W}_\ell\}_\ell} \text{CE}(f; \{\mathbf{W}_\ell\}_\ell) + \lambda \sum_\ell \left\|\mathbf{W}_\ell\right\|_2 \times \left\|\mathbf{W}_\ell^{-1}\right\|_2. \quad (6)$$

While this formulation is well grounded and specifically tailored to our goal (i.e., learning stable bidirectional networks), the optimization of condition number presents a significant challenge primarily due to its non-convexity and non-smoothness, rendering traditional optimization techniques (such as gradient descent) difficult. Furthermore, the condition number's dependence on eigenvalues, as nonlinear measures of matrices, makes gradients estimation unstable and optimization challenging especially for large-scale matrices. Besides, striking a balance between cross-entropy and condition number minimization makes the problem even harder (see later performances in tables 5-6). In what follows, we consider a surrogate term that *formally* has optima with unitary condition numbers —similarly to the regularizer in Eq. 6— while making optimization more tractable in practice; thereby exhibiting better performances (as shown later in experiments). Hence, instead of minimizing directly the condition number in the loss, we constrain the matrices in θ to be *orthonormal* which also guarantees their invertibility. With this update, the global loss, when training GCNs, becomes

$$\min_{\{\mathbf{W}_\ell\}_\ell} \text{CE}(f; \{\mathbf{W}_\ell\}_\ell) + \lambda \sum_\ell \left\|\mathbf{W}_\ell^\top \mathbf{W}_\ell - \mathbf{I}\right\|_F, \quad (7)$$

here $\mathbf{I}$ stands for identity, $\|.\|_F$ denotes the Frobenius norm and $\lambda > 0$ (with $\lambda = \frac{1}{p}$ in practice[3]); in particular, when $\mathbf{W}_\ell^\top \mathbf{W}_\ell - \mathbf{I} = 0$, then $\mathbf{W}_\ell^{-1} = \mathbf{W}_\ell^\top$ and $\|\mathbf{W}_\ell\|_2 = \|\mathbf{W}_\ell^{-1}\|_2 = 1$, so $\kappa = 1$ and the KM-Lipschitz constant in Eq. 5 becomes tighter. With this updated loss, the learned GCNs are guaranteed to be invertible and stable while also being discriminative as shown later in experiments.

3.3. Weight reparametrization

In order to further enhance the stability of the learned network f, we consider a *weight reparametrization* (WR) as $\{\mathbf{W}_\ell = \hat{\mathbf{W}}_\ell + \delta\mathbf{I}\}_\ell$, with $\delta \geq 0$. This transformation ensures that the eigenvalues of $\mathbf{W}_\ell$ — given by $\{\delta_i + \delta\}_i$, with $\{\delta_i\}_i$ the eigenvalues of $\hat{\mathbf{W}}_\ell$ — are bounded below by δ. Therefore, the condition number of $\mathbf{W}_\ell$ is further reduced to $\max_i |\delta_i + \delta| \times \max_i |1/(\delta_i + \delta)|$. A lower condition number signifies that any slight update of exemplars in the latent space (with the fixed-point iteration in Eq. 2) will also result into a slight update of these exemplars in the ambient (input) space when applying f^{-1}. Conversely, this also guarantees that slight updates of data in the ambient space will also results into stable responses when applying f. Nonetheless, achieving an optimal condition number (approaching unity) — without overestimating δ and compromising the expressiveness of the learned networks — remains challenging when using only this reparametrization. Thus, an explicit regularization of the cross-entropy loss, as shown in Eqs. 6 and 7, is also necessary in order to circumvent the need for overestimated δ values (see later tables 5-6). Note that, with this WR, the gradient of the loss in Eqs. 6-7 w.r.t. $\hat{\mathbf{W}}$, denoted as $\nabla_{\hat{\mathbf{W}}}\mathcal{L}$, remains identical to $\nabla_{\mathbf{W}}\mathcal{L}$ as $\nabla_{\hat{\mathbf{W}}}\mathcal{L} = \nabla_{\mathbf{W}}\mathcal{L} \cdot \frac{\partial \mathbf{W}}{\partial \hat{\mathbf{W}}}$ (chain rule), and $\frac{\partial \mathbf{W}}{\partial \hat{\mathbf{W}}}$ is simply the identity matrix (as $\mathbf{W} = \hat{\mathbf{W}} + \delta\mathbf{I}$). Hence, this WR directly shifts the eigenvalues, and further improves stability, without changing the gradient of the loss.

4. Experiments

This section presents an evaluation of the performance of baseline and our label-frugal GCNs for skeleton-based action recognition. The evaluation is conducted using the SBU Interaction [58] and First Person Hand Action (FPHA) [11] datasets. The SBU Interaction dataset, acquired via Microsoft Kinect, comprises 282 skeleton sequences

[2]This setting guarantees a small ratio between u, l, and contributes in making the KM constant $(\kappa\, u/l)^{L-1}$ small, depending also on the condition number κ (see again proposition 2).

[3]Note that at frugal data regimes, the cross entropy term involves few labeled data, so it is enough to set λ to small values in order to guarantee the minimization of both terms.

representing dyadic interactions. These interactions are categorized into eight predefined action classes. Each sequence consists of two 15-joint skeletons, with each joint represented by its 3D spatial coordinates across the temporal domain. Evaluation adheres to the established train-test partitioning specified in [58]. The FPHA dataset encompasses 1175 skeleton sequences, covering 45 diverse hand-action categories. These actions are performed by six subjects across three distinct scenarios, exhibiting significant intra-class variability in style, velocity, scale, and viewpoint. Each skeleton sequence represents 21 hand joints, with each joint's 3D coordinates taken over time. Following the evaluation protocol defined in [11], a 1:1 train-test split is employed, with 600 sequences allocated for training and 575 for testing. For both datasets, we report the average classification accuracy across all action categories.

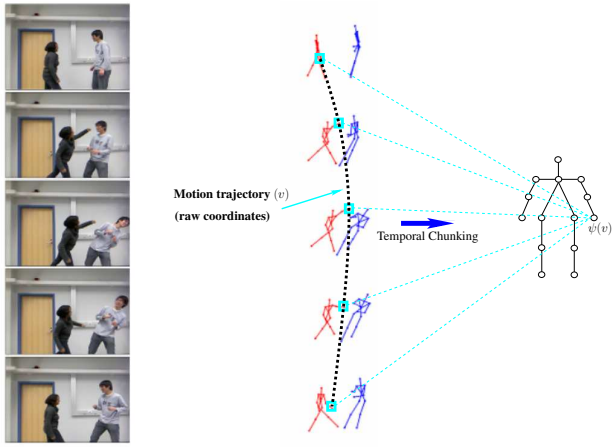

Figure 2. This figure shows the whole keypoint tracking and description process.

Method	Accuracy (%)
Raw Position [58]	49.7
Joint feature [17]	86.9
CHARM [26]	86.9
H-RNN [7]	80.4
ST-LSTM [27]	88.6
Co-occurrence-LSTM [63]	90.4
STA-LSTM [46]	91.5
ST-LSTM + Trust Gate [27]	93.3
VA-LSTM [60]	97.6
GCA-LSTM [28]	94.9
Riemannian manifold. traj [21]	93.7
DeepGRU [30]	95.7
RHCN + ACSC + STUFE [25]	98.7
Our baseline GCN	98.4

Table 1. Comparison of our baseline GCN (not label-efficient) against related work on the SBU database.

Method	Color	Depth	Pose	Accuracy (%)
2-stream-color [9]	✓	✗	✗	61.56
2-stream-flow [9]	✓	✗	✗	69.91
2-stream-all [9]	✓	✗	✗	75.30
HOG2-dep [34]	✗	✓	✗	59.83
HOG2-dep+pose [34]	✗	✓	✓	66.78
HON4D [35]	✗	✓	✗	70.61
Novel View [38]	✗	✓	✗	69.21
1-layer LSTM [63]	✗	✗	✓	78.73
2-layer LSTM [63]	✗	✗	✓	80.14
Moving Pose [59]	✗	✗	✓	56.34
Lie Group [47]	✗	✗	✓	82.69
HBRNN [7]	✗	✗	✓	77.40
Gram Matrix [62]	✗	✗	✓	85.39
TF [11]	✗	✗	✓	80.69
JOULE-color [14]	✓	✗	✗	66.78
JOULE-depth [14]	✗	✓	✗	60.17
JOULE-pose [14]	✗	✗	✓	74.60
JOULE-all [14]	✓	✓	✓	78.78
Huang et al. [15]	✗	✗	✓	84.35
Huang et al. [16]	✗	✗	✓	77.57
HAN [29]	✗	✗	✓	85.74
Our baseline GCN	✗	✗	✓	88.17

Table 2. Comparison of our baseline GCN (not label-efficient) against related work on the FPHA database.

Input graphs. Each skeleton sequence, denoted as $\{S_t\}_{t=1}^{T}$, is represented as a temporal series of 3D joint coordinates, $S_t = \{\hat{p}_{tj}\}_{j=1}^{J}$, where T denotes the sequence length and J the number of joints. The trajectory of a joint j, $\{\hat{p}_{tj}\}_{t=1}^{T}$, describes its spatial displacement over time. A graph representation, referred to as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is defined where $\mathcal{V}$ corresponds to nodes $v_j \in \mathcal{V}$, each one representing a joint trajectory $\{\hat{p}_{tj}\}_{t=1}^{T}$. The set $\mathcal{E}$ contains edges $(v_j, v_i) \in \mathcal{E}$ connecting spatially adjacent joint trajectories. To facilitate temporal processing, each joint trajectory is partitioned into M_c equal temporal intervals (chunks), with $M_c = 4$ in practice. Joint coordinates $\{\hat{p}_{tj}\}_{t=1}^{T}$ are assigned to these intervals based on their temporal indices. The mean 3D coordinates within each interval is computed, and these coordinates are concatenated to form a trajectory descriptor $\psi(v_j) \in \mathbb{R}^s$ of dimensionality $s = 3M_c$ (see Fig. 2). This temporal chunking effectively encodes the temporal dynamics while ensuring invariance to frame rate and sequence duration.

Implementation details & baseline GCNs. All GCN models are trained using the Adam optimizer for 2700 epochs. The training batch size is set to 200 for the SBU Interaction dataset and 600 for the FPHA dataset. A momentum parameter of 0.9 is used. The global learning rate ν is dynamically adjusted based on the temporal derivative of the loss function, as defined in Eqs. 6-7. Specifically, ν is scaled by a factor of 0.99 when the temporal derivative of the loss function increases, and by a factor of 1/0.99 otherwise, im-

plementing an adaptive learning rate strategy. Training has been conducted on a GeForce GTX 1070 GPU with 8 GB of memory. No dropout regularization or data augmentation are employed. For the SBU Interaction dataset, the GCN architecture consists of three sequential layers, each comprising a mono-head attention mechanism followed by a convolutional layer with 8 filters. This is succeeded by a fully connected layer and a classification layer. For the FPHA dataset, a comparatively larger GCN architecture is used, differing from the SBU architecture primarily in the number of convolutional filters, employing 16 filters instead of 8. Both GCN architectures, when evaluated on the SBU Interaction and FPHA benchmarks, demonstrate high classification accuracy as detailed in Tables 1 and 2. Subsequently, the objective is to achieve label-efficient learning while maintaining performance as close as possible to the baseline accuracy.

Labeling rates	Accuracy	Observation
100%	98.40	Baseline GCN (not label-efficient)
45%	<u>89.23</u>	wo display model (random display)
	<u>89.23</u>	+ display model + ambient (our)
	93.84	+ display model + latent (our)
	67.69	uncertainty (margin-based)
	83.07	diversity (coreset-based)
30%	80.00	wo display model (random display)
	<u>86.15</u>	+ display model + ambient (our)
	87.69	+ display model + latent (our)
	61.53	uncertainty (margin-based)
	83.07	diversity (coreset-based)
15%	<u>69.23</u>	wo display model (random display)
	75.38	+ display model + ambient (our)
	75.38	+ display model + latent (our)
	56.92	uncertainty (margin-based)
	66.15	diversity (coreset-based)

Table 3. This table shows detailed performances and ablation study on SBU for different labeling rates. Here "wo" stands for "without". Best results are shown in bold and second best results underlined.

4.1. Display model: comparison & ablation

Tables 3-4 present a comparative analysis and ablation study of the proposed method on the SBU and FPHA datasets, respectively. The results demonstrate that the application of our display model within the ambient space yields high classification accuracy, often surpassing comparative display selection strategies by a significant margin. Furthermore, the use of the latent space results in a further performance improvement, highlighting the efficacy of our model and its synergy with latent space representations. Comparative analysis against alternative display selection strategies, including random sampling, diversity-based [22] and uncertainty-based selection [53], all integrated with our GCN learning framework, reveals a substantial performance gain achieved by our method. As showcased in Tables 3-4, our method exhibits significant performance advantages across various equivalent labeling rates. The results also indicate that random sampling achieves competitive performance, particularly at higher sampling rates (e.g., 45%), consistent with prior observations (e.g., [43]). However, at lower sampling rates (e.g., 15%), the performance of random sampling diminishes, necessitating more elaborate selection strategies. While uncertainty-based selection refines classifications, it lacks sufficient diversity. Random and diversity-based selection methods, conversely, fail to adequately refine classifications. Moreover, all comparative methods suffer from the inherent rigidity of selecting displays from a fixed pool. In contrast, our display model learns adaptable exemplars constrained within the latent space of the proposed stable and invertible bidirectional GCNs, resulting in improved performance, especially under frugal labeling regimes. This adaptability allows for a more effective representation of the data, leading to enhanced classification accuracy.

Labeling rates	Accuracy	Observation
100%	88.17	Baseline GCN (not label-efficient)
45%	<u>75.47</u>	wo display model (random display)
	72.52	+ display model + ambient (our)
	75.65	+ display model + latent (our)
	63.30	uncertainty (margin-based)
	70.26	diversity (coreset-based)
30%	**67.47**	wo display model (random display)
	61.21	+ display model + ambient (our)
	<u>63.65</u>	+ display model + latent (our)
	56.17	uncertainty (margin-based)
	62.08	diversity (coreset-based)
15%	40.52	wo display model (random display)
	45.21	+ display model + ambient (our)
	49.21	+ display model + latent (our)
	41.73	uncertainty (margin-based)
	<u>46.26</u>	diversity (coreset-based)

Table 4. This table shows detailed performances and ablation study on FPHA for different labeling rates. Here "wo" stands for "without". Best results are shown in bold and second best results underlined.

4.2. Regularization and weight reparametrization

Tables 5-6 show an analysis of the individual and combined effects of our used regularizers —– namely, Condition Number (CN) and Orthogonality Regularization (OR)—– and WR. The observed results demonstrate a consistent positive impact of WR, both individually and in conjunction with regularization. Notably, with the exception of OR regularization (configs #7,#8), WR significantly reduces both the observed CN and Fréchet Inception Distance (FID), especially when δ is sufficiently large, while concurrently improving classification accuracy relative to the non-reparametrized baseline (configs #2,#3,#4 vs #1 and #6 vs #5). This behavior is observed across a range of δ

Regularizer	WR ($\mathbf{W}+\delta I$)	Acc ↑	Observed CN ↓	FID Score ↓	config
No	No	9.23	1.85×10^{29}	6.44×10^{15}	#1
No	Yes, $\delta = 10^6$	58.46	2.022	**7.16**	#2
No	Yes, $\delta = 10^5$	<u>83.07</u>	154.52	8.88	#3
No	Yes, $\delta = 10^1$	<u>83.07</u>	5.01×10^{11}	92.04	#4
CN	No	9.23	3×10^9	3973.2	#5
CN	Yes, $\delta = 10^1$	44.23	<u>1.015</u>	15.85	#6
OR	No	**93.84**	5.410	10.18	#7
OR	Yes, $\delta = 10^1$	81.53	**1.010**	<u>8.70</u>	#8

Table 5. This table shows the impact of different regularizers (OR and CN) and WR (for different setting of δ) when taken individually and combined. Here Acc (accuracy), observed CN and FID scores are shown on the SBU dataset. Best results are shown in bold and second best results underlined.

Regularizer	WR ($\mathbf{W}+\delta I$)	Acc ↑	Observed CN ↓	FID Score ↓	config
No	No	54.78	2.91×10^{22}	5.30×10^9	#1
No	Yes, $\delta = 10^6$	2.26	4.666	6.32	#2
No	Yes, $\delta = 10^5$	54.78	32.362	5.87	#3
No	Yes, $\delta = 10^1$	57.04	1.19×10^{11}	13.33	#4
CN	No	2.08	2.89×10^{30}	1.86×10^{12}	#5
CN	Yes, $\delta = 10^1$	64.17	**1.000**	7.05	#6
OR	No	**75.65**	<u>1.052</u>	**2.37**	#7
OR	Yes, $\delta = 10^1$	<u>68.34</u>	1.055	<u>5.54</u>	#8

Table 6. This table shows the impact of different regularizers (OR and CN) and WR (for different setting of δ) when taken individually and combined. Here Acc (accuracy), observed CN and FID scores are shown on the FPHA dataset. Best results are shown in bold and second best results underlined.

values. An overestimated δ (config #2) imposes excessive rigidity, resulting in minimal FID and CN values. However, this rigidity impedes the network's ability to minimize cross-entropy, thereby compromising classification accuracy. Conversely, an underestimated δ (config #4) grants higher model flexibility, facilitating effective cross-entropy minimization. Nevertheless, this leads to limited generalization, evidenced by elevated FID and CN scores, indicating out-of-distribution exemplars. An intermediate δ value (config #3) achieves a more favorable balance, optimizing the efficacy of reparametrization. When combined with CN regularization (config #6), the reparametrization exhibits reduced dependency on large δ values, and effectively mitigates FID and CN, diminishing the criticality of precise δ tuning with large values. Consequently, the selection of δ becomes easier. Across all experimental results, OR (configs #7,#8) provides a consistent and notable improvement in accuracy, FID and observed CN, with or without reparametrization. This confirms the effectiveness of OR as a stronger regularizer against CN.

5. Conclusion

This paper introduces a label-efficient method for skeleton-based action recognition using graph convolutional networks (GCNs). By minimizing the need for extensive labeled data, this approach enhances the practicality of GCNs in scenarios with limited annotation. The primary contribution of this work lies in the design of a new acquisition function as the solution of an objective function. This function balances representativity, diversity, and uncertainty, yielding a solution that optimally reflects the underlying data distribution. Furthermore, we upgrade our design by making our GCNs bidirectional and stable thereby yielding learned latent spaces with enhanced representational and discriminative power. Extensive experiments on two challenging skeleton-based recognition datasets validate the efficacy and superior performance of our method.

Appendix

Sketch of the Proof (Proposition 2). Given a metric space (A, d_A), where d_A denotes the metric on the set A (by default d_A is taken as ℓ_2 and A as $\mathbb{R}^p$); considering a subsumed version of our GCNs, and using the Lipschitz continuity, one may write $d_A(f(\phi_1^1), f(\phi_2^1)) = (*)$ with

$$\begin{aligned}(*) &= d_A(g_L(\mathbf{W}_L^\top \phi_1^{L-1}), g_L(\mathbf{W}_L^\top \phi_2^{L-1})) \\ &\leq u\, d_A(\mathbf{W}_L^\top \phi_1^{L-1}, \mathbf{W}_L^\top \phi_2^{L-1}) \\ &\leq u.\|\mathbf{W}_L\|_A \;\; d_A(\phi_1^{L-1}, \phi_2^{L-1}) \\ &\leq u^{L-1}\|\mathbf{W}_L\|_A \ldots \|\mathbf{W}_2\|_A \;\; d_A(\phi_1^1, \phi_2^1),\end{aligned}$$

being ϕ_1^1, ϕ_2^1 two network inputs. From above inequality, it follows that $d_A(f(\phi_1^1), f(\phi_2^1)) \leq K\; d_A(\phi_1^1, \phi_2^1)$ with $K = u^{L-1}\prod_\ell \|\mathbf{W}_\ell\|_A$. Similarly for f^{-1}, given an output ϕ^L, $f^{-1}(\phi^L) = \phi^1$ with $\phi^{\ell-1} = (\mathbf{W}_\ell^\top)^{-1} g_\ell^{-1}(\phi^\ell)$; considering two network outputs ϕ_1^L, ϕ_2^L, one may write $d_A(f^{-1}(\phi_1^L), f^{-1}(\phi_2^L)) = (*)$ with

$$\begin{aligned}(*) = \;& d_A((\mathbf{W}_2^\top)^{-1} g_2^{-1}(\phi_1^2), (\mathbf{W}_2^\top)^{-1} g_2^{-1}(\phi_2^2)) \\ \leq \;& \|(\mathbf{W}_2^\top)^{-1}\|_A \;\; d_A(g_2^{-1}(\phi_1^2), g_2^{-1}(\phi_2^2)) \\ \leq \;& \|(\mathbf{W}_2^\top)^{-1}\|_A \; (1/l)\; d_A(\phi_1^2, \phi_2^2) \\ \leq \;& \textstyle\prod_\ell \|(\mathbf{W}_\ell^\top)^{-1}\|_A \; (1/l)^{L-1} \; d_A(\phi_1^L, \phi_2^L).\end{aligned}$$

It follows that $d_A(f^{-1}(\phi_1^L), f^{-1}(\phi_2^L)) \leq M d_A(\phi_1^L, \phi_2^L)$ with $M = (1/l)^{L-1}\prod_\ell \|(\mathbf{W}_\ell^\top)^{-1}\|_A$. Now by combing K and M, the KM-Lipschitz constant can be obtained as

$$\begin{aligned}KM &= (u/l)^{L-1}\,(1/l)^{L-1} \prod_{\ell=1}^{L-1} \|(\mathbf{W}_\ell)\|_A \|\mathbf{W}_\ell^{-1}\|_A \\ &\leq (\kappa\, u/l)^{L-1}.\end{aligned}$$

□

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020.

[2] Erdem Bıyık, Kenneth Wang, Nima Anari, and Dorsa Sadigh. Batch active learning using determinantal point processes. 2019.

[3] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. Active learning for deep object detection. *arXiv preprint arXiv:1809.09875*, 2018.

[4] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th international conference on data mining*, pages 51–60. IEEE, 2013.

[5] Razvan Caramalau, Binod Bhattarai, Danail Stoyanov, and Tae-Kyun Kim. Mobyv2al: Self-supervised active learning for image classification. 2023.

[6] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, pages 746–751, 2005.

[7] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[8] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. 2017.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[11] Guillermo Garcia-Hernando and Tae-Kyun Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 432–440, 2017.

[12] Juha Heinonen. *Lectures on Lipschitz analysis*. Number 100. University of Jyväskylä, 2005.

[13] Patrick Hemmer, Niklas Kühl, and Jakob Schöffer. Deal: Deep evidential active learning for image classification. pages 171–192. Springer, 2022.

[14] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5344–5352, 2015.

[15] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[16] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[17] Yanli Ji, Guo Ye, and Hong Cheng. Interactive body part contrast mining for human interaction recognition. In *2014 IEEE international conference on multimedia and expo workshops (ICMEW)*, pages 1–6. IEEE, 2014.

[18] M. Jiu and H. Sahbi. Nonlinear deep kernel learning for image annotation. *IEEE Transactions on Image Processing*, 26 (4):1820–1832, 2017.

[19] M. Jiu and H. Sahbi. Deep representation design from deep kernel networks. *Pattern Recognition*, 88:447–457, 2019.

[20] Seohyeon Jung, Sanghyun Kim, and Juho Lee. A simple yet powerful deep active learning with snapshots ensembles. In *The Eleventh International Conference on Learning Representations*, 2022.

[21] Anis Kacem, Mohamed Daoudi, Boulbaba Ben Amor, Stefano Berretti, and Juan Carlos Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):1–14, 2018.

[22] Yeachan Kim and Bonggun Shin. In defense of core-set: A density-aware core-set selection for active learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 804–812, 2022.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[24] Seong Min Kye, Kwanghee Choi, Hyeongmin Byun, and Buru Chang. Tidal: Learning training dynamics for active learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22335–22345, 2023.

[25] Sheng Li, Tingting Jiang, Tiejun Huang, and Yonghong Tian. Global co-occurrence feature learning and active coordinate system conversion for skeleton-based action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 586–594, 2020.

[26] Wenbo Li, Longyin Wen, Mooi Choo Chuah, and Siwei Lyu. Category-blind human action recognition: A practical recognition system. In *Proceedings of the IEEE international conference on computer vision*, pages 4444–4452, 2015.

[27] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, pages 816–833. Springer, 2016.

[28] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017.

[29] Jianbo Liu, Ying Wang, Shiming Xiang, and Chunhong Pan. Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition. *arXiv preprint arXiv:2106.13391*, 2021.

[30] Mehran Maghoumi and Joseph J LaViola. Deepgru: Deep gesture recognition utility. In *Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14*, pages 16–31. Springer, 2019.

[31] Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. 2023.

[32] A. Mazari and H. Sahbi. Mlgcn: Multi-laplacian graph convolutional networks for human action recognition. In *The British machine vision conference (BMVC)*, 2019.

[33] A. Mazari and H. Sahbi. Deep multiple aggregation networks for action recognition. *International Journal of Multimedia Information Retrieval*, 13(1):9, 2024.

[34] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[35] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, 2013.

[36] Jaehyun Park, Dongmin Park, and Jae-Gil Lee. Active learning for continual learning: Keeping the past alive in the present. 2025.

[37] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. 2022.

[38] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016.

[39] H. Sahbi. Learning laplacians in chebyshev graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2064–2075, 2021.

[40] H. Sahbi. Learning connectivity with graph convolutional networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9996–10003. IEEE, 2021.

[41] H. Sahbi, J-Y. Audibert, and R. Keriven. Context-dependent kernels for object classification. *IEEE transactions on pattern analysis and machine intelligence*, 33(4):699–708, 2010.

[42] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. 2017.

[43] Burr Settles. Active learning literature survey. Technical report, 2009.

[44] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. pages 287–294, 1992.

[45] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[46] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[47] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[48] Keze Wang, Liang Lin, Xiaopeng Yan, Ziliang Chen, Dongyu Zhang, and Lei Zhang. Cost-effective object detection: Active sample mining with switchable selection criteria. *IEEE transactions on neural networks and learning systems*, 30(3):834–850, 2018.

[49] L. Wang and H. Sahbi. Directed acyclic graph kernels for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3168–3175, 2013.

[50] L. Wang and H. Sahbi. Nonlinear cross-view sample enrichment for action recognition. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13*, pages 47–62. Springer, 2015.

[51] Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active learning for graph neural networks via node feature propagation. 2019.

[52] Yong Cheng Wu. Active learning based on diversity maximization. *Applied Mechanics and Materials*, 347:2548–2552, 2013.

[53] Zongyi Xu, Bo Yuan, Shanshan Zhao, Qianni Zhang, and Xinbo Gao. Hierarchical point-based active learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18098–18108, 2023.

[54] Xuyang Yan, Shabnam Nazmi, Biniam Gebru, Mohd Anwar, Abdollah Homaifar, Mrinmoy Sarkar, and Kishor Datta Gupta. A clustering-based active learning method to query informative and representative samples. pages 13250–13267. Springer, 2022.

[55] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. pages 22354–22367, 2022.

[56] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

[57] F. Yuan, G-S. Xia, H. Sahbi, and V. Prinet. Mid-level features and spatio-temporal context for activity recognition. *Pattern Recognition*, 45(12):4182–4191, 2012.

[58] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 28–35. IEEE, 2012.

[59] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2752–2759, 2013.

[60] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recog-

nition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pages 2117–2126, 2017.

[61] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 148–157. IEEE, 2017.

[62] Xikang Zhang, Yin Wang, Mengran Gou, Mario Sznaier, and Octavia Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4498–4507, 2016.

[63] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2016.