

Constrained Video Face Clustering using 1NN Relations

Vicky Kalogeiton
vicky@robots.ox.ac.uk

Andrew Zisserman
az@robots.ox.ac.uk

Visual Geometry Group
University of Oxford
<http://www.robots.ox.ac.uk/~vgg/research/c1c>

Abstract

In this work, we introduce the Constrained first nearest neighbour Clustering (C1C) method for video face clustering. Using the premise that the first nearest neighbour (1NN) of an instance is sufficient to discover large chains and groupings, C1C builds upon the hierarchical clustering method FINCH by imposing must-link and cannot-link constraints acquired in a self-supervised manner. We show that adding these constraints leads to performance improvements with low computational cost. C1C is easily scalable and does not require any training. Additionally, we introduce a new *Friends* dataset for evaluating the performance of face clustering algorithms. Given that most video datasets for face clustering are saturated or emphasize only the main characters, the *Friends* dataset is larger, contains identities for several main and secondary characters, and tackles more challenging cases as it labels also the ‘back of the head’. We evaluate C1C on the Big Bang Theory, Buffy, and Sherlock datasets for video face clustering, and show that it achieves the new state of the art whilst setting the baseline on *Friends*.

1 Introduction

Detecting and identifying characters in movies and TV shows is a key element for story understanding [57]. Clustering characters by identity is one step towards this, as it can reduce the tremendous annotation time and cost. A successful solution to video character clustering can have a significant impact on various tasks, such as browsing and organization of movie collections, and even automatic collection of large-scale TV show datasets.

The problem of video face clustering has been tackled since the early 2000s [8, 15, 40, 47]. At first glance, deep learning seems to have solved the problem, as there exist methods with near perfect performance [36, 41]. However, the current set of evaluation datasets is limited, for example they only consider the principal characters and ignore the secondary or background characters [14, 29, 31], and this is hiding some of the shortcomings of current clustering methods [34, 35, 36, 54]. To address these dataset limitations we introduce a new *Friends* dataset that contains approximately 18k annotated heads for 49 characters (Figure 2). Compared to previous datasets, *Friends* is larger, contains many secondary characters, and is more challenging as it also contains back of the head detections.

Furthermore, we propose a new video face clustering method: Constrained 1NN hierarchical Clustering method (C1C). C1C combines the clustering method FINCH [32] with

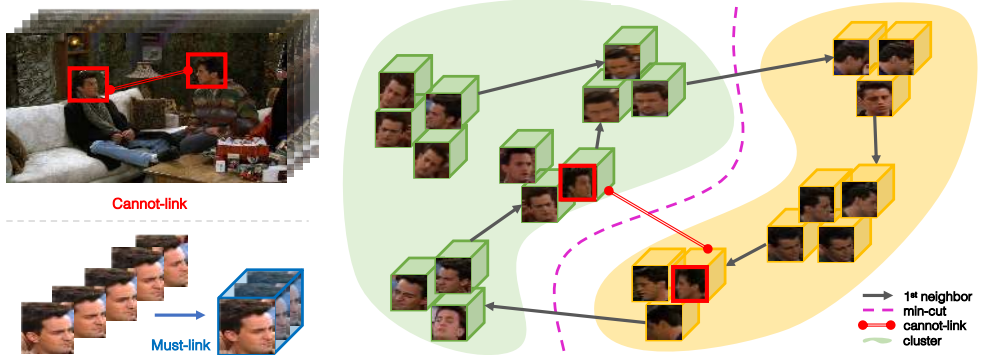


Figure 1: **Constrained video face clustering.** (Left) Must-link constraint: instances from the same track must be linked as they represent the same character. Cannot-link constraint: concurrent tracks appearing in the same frame cannot be linked, as they represent different characters. (Right) C1C Clustering given first NN relations and constraints.

must-link and cannot-link constraints [8, 47]. FINCH [32] is computationally efficient as it only relies on first neighbor relations and it performs well on face clustering [36]. The imposed constraints [8, 47] are acquired in a self-supervised manner: instances from the same track *must* be linked as they represent the same character, while concurrent tracks *cannot* be linked, as they represent different characters (Figure 1). Unlike other hierarchical clustering methods, the cannot-link constraints offer a natural lower bound on the resulting number of clusters, as well as reducing the search space, and therefore the computational time.

Our experiments show that C1C achieves state-of-the-art results on the The Big Bang Theory, Buffy, and Sherlock datasets [14, 27, 31]. Note that most methods for video face clustering [34, 35, 36, 41] require training on the same domain as the one on which they are evaluated, i.e. the same TV show or even the same episode. In practice, however, this is a bottleneck, as it requires annotated data for every different movie or TV show. In contrast, we use a face representation for each track from a pre-trained and fixed CNN, without any knowledge of the domain or the test TV show and we outperform the state of the art.

In summary, we make the following contributions: (i) we introduce the Friends dataset for character clustering in videos – three times larger than the biggest available benchmark and tackling more difficult cases; (ii) we propose C1C, a hierarchical method for video face clustering that combines the clustering method FINCH [32] with must-link and cannot-link constraints; and (iii) we demonstrate state-of-the-art results on existing datasets and report the first results on Friends revealing the importance of new methods.

2 Related work

Face clustering is an extensively studied task [4, 16, 18, 22, 28, 37, 53]. Most representative methods are based on graphs, e.g. ARO [28] predicts if a node should be linked to its kNN computed using Approximate Nearest Neighbors (ANN), while [37] also relies on ANN to scale the proposed algorithm to more data. Recent approaches use graph convolutional networks (GCN) [45, 50]. [52] maximizes the mutual information between global and local graph representations, while others cast the problem as link prediction [45] or

subgraphs [51]. [50] combine a detection and a segmentation module to pinpoint clusters.

Constrained Clustering typically incorporates a set of must and/or cannot-link constraints with a clustering algorithm [11, 43]. Given the temporal continuity of videos, must and cannot-link constraints have also been adopted by the vision community for various tasks, such as video face recognition [24, 49] and clustering [8, 39]. For instance, [47] uses pairwise constraints together with label-level and constraint-level local smoothness, [8] uses them to learn cast-specific metrics, while [3] combines them with weakly labeled data. These works have shown that adding them leads to performance improvements. C1C iteratively imposes constraints while grouping INN, showing performance improvements.

Video face clustering has been an active research topic for several years [8, 15, 40, 47]. Early methods rely on handcrafted face features and the temporal continuity of videos. [8] uses metric learning with automatically obtained positive and negative face pairs to learn cast-specific distances. [46, 47] iteratively clusters and associates face tracklets based on Hidden Markov Random Field, whereas WBSLRR [48] considers the prior knowledge while learning a weighted block-sparse low rank representation. Recent methods typically rely on face features coming from powerful CNNs [33, 36, 40, 41]. For instance, the Siamese-based TSiam and SSiam [34, 35] methods mine positive and negative pairs by sorting distances (SSiam) for singleton tracks (TSiam). [19] propose a rank-1 count similarity method for joint face detection and clustering. The hierarchical clustering method FINCH [32] links samples through the first NN relations. Given several pure face tracks from FINCH [32], CCL [36] forms positive and negative pairs used as pseudo-labels to train a MLP with a contrastive loss. In a similar setting [41] creates a fixed-radius embedding for each character to be clustered. Except for [32], all methods require training on the test domain to obtain face embeddings and are based on a hierarchical agglomerative clustering (HAC) to produce the final clusters. In contrast, C1C does not require any training as the must-link and cannot-link constraints are incorporated in the clustering method.

Bias is an issue in both face datasets and face models [2, 10, 38]. Some works have tried to address it [44]; for instance, [17] investigates the gender-bias generation, [9] tries to learn it, while [1] aims at removing it from feature representations. Here, we use pre-trained face embeddings [5, 7] and cluster the features accordingly. These detectors may have inherited potential bias from the corresponding datasets they were trained on. We argue, however, that using the imposed constraints helps alleviate possible bias in the clustering results.

3 C1C: Constrained INN Clustering

The goal of this work is to cluster video face identities using video-level constraints. To this end, we introduce the **Constrained INN Clustering** method. C1C combines an existing hierarchical clustering algorithm (FINCH [32]) with the idea of imposing constraints naturally occurring in videos in a self-supervised manner [8, 39]. Specifically, C1C groups instances that share a first NN, as long as they do not violate some must-link and cannot-link constraints acquired in a self-supervised manner (Figure 1). Here, first we briefly describe FINCH, and then, we describe C1C and provide a discussion of our findings (Section 3.1). Algorithm 1 states the steps of FINCH with the additions (**bold**) leading to C1C.

Hierarchical clustering methods. We work with hierarchical methods, as they typically find a good local minimum solution with reasonable complexity. For face clustering we prefer bottom-up methods as they make clustering decisions based on local patterns, without taking into account the global distribution. Furthermore, they are appropriate here, where the face of one character is more similar within smaller temporal windows – within the same episode characters look more alike than across episodes. Note that the imposed self-supervised constraints can be applied in most hierarchical bottom-up clustering methods.

FINCH Clustering method [32]. First Integer Neighbour indices producing a Clustering Hierarchy (FINCH) is a fast and scalable method. Unlike standard HAC methods where each partition merges only one instance with existing clusters, it groups several instances using first NN relations at the same partition. In the first partition, it links samples through first NN relations, while all following partitions link the clusters from the previous step. Given pairwise distances between all instances, it considers only the first NN $n_{x_i}^1$ of each instance $x_i \in \mathbb{R}^d$. This results in cases where the first neighbour is not mutual between two instances x_i and x_j . Then, at every partition Γ , it forms K_Γ clusters by merging instances that either are first neighbours or have a common first neighbour, as described by the adjacency matrix:

$$A(i, j) = \begin{cases} 1 & \text{if } x_j = n_{x_i}^1 \text{ or } n_{x_j}^1 = x_i \text{ or } n_{x_i}^1 = n_{x_j}^1, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

By representing each formed cluster with only one instance, FINCH recursively merges clusters until one cluster remains. At every partition, instead of computing similarities between each instance of each cluster, it computes the mean of a cluster $m_\Gamma^k \in \mathbb{R}^d$, for $k = 1, \dots, K_\Gamma$ and uses this mean to compute its pairwise similarity to the means of other clusters.

3.1 The proposed C1C method

The constraints. Instance level constraints can express a priori knowledge about which instances should be grouped together or not [43]. Videos provide this knowledge without any labelling or supervision with two types of constraints: (i) must-link constraints specify that two instances have to be in the same cluster, i.e. faces in the same track must depict the same character, and (ii) cannot-link constraints specify that two instances must not be placed in the same cluster, i.e. tracks that temporally overlap must depict different characters.

Description. C1C first computes must-link and cannot-link constraints for all instances x in a self-supervised manner. The must-link constraints define a transitive binary relation over the instances [43, 47]. Consequently, by using both constraints, we take a transitive closure over the constraints¹. In practice, enforcing this is computationally expensive, as it requires computing all possible combinations of groupings subject to the constraints. To this end, C1C first groups instances into clusters based on Equation 1 (step 6 in Algorithm 1), and then enforces the constraints. Since only the must-link constraints are transitive, it is possible for a candidate cluster to contain cannot-link constraints. In such cases, C1C iteratively splits the violating cluster using the max-flow min-cut algorithm until no constraint is violated (Figure 1(right), steps 7-13 in Algorithm 1).

¹If x_i must link to x_j that cannot link to x_k , then we know that x_i cannot link to x_k .

Algorithm 1 Proposed C1C algorithm

-
- Input** instances $x_i \in \mathbb{R}^d$ for $i \in [1, \dots, M]$
Output L partitions $\{C_\Gamma^1, \dots, C_\Gamma^{K_\Gamma}\}$ of x_i for $\Gamma \in [1, \dots, L]$
- 1: **Compute must-link** $\mathcal{M} \subset [1, N]^2$ **and cannot-link** $\mathcal{C} \subset [1, N]^2$ **constraints**
 - 2: Set the clusters $C_0^k \subset \mathbb{R}^d$ of partition $\Gamma = 0$ to be the connected components formed by the must-link constraints \mathcal{M}
 - 3: **while** there are at least two clusters in partition Γ **do**
 - 4: For $k = 1 \rightarrow K_\Gamma$, compute the mean $m_\Gamma^k \in \mathbb{R}^d$ of the cluster C_Γ^k
 - 5: For $k = 1 \rightarrow K_\Gamma$, compute the first neighbor of m_Γ^k **that does not violate a cannot-link constraint in** \mathcal{C}
 - 6: Apply Eq. 1 to the mean m_Γ^k to build the clusters $C_{\Gamma+1}^k$ of instances $\{x_i\}$
 - 7: **while** \exists a **cannot-link violation** $(k_1, k_2) \in \mathcal{C}$ **in a cluster** $C_{\Gamma+1}^k$ **do**
 - 8: **Split cluster** $C_{\Gamma+1}^k$ **into 2 clusters separating** k_1 **and** k_2
 - 9: $K_{\Gamma+1} \leftarrow K_{\Gamma+1} + 1$
 - 10: **end while**
 - 11: **Update cannot-link constraints** $\mathcal{C} \leftarrow \{(k'_1, k'_2) | (k_1, k_2) \in \mathcal{C}\}$ **where** k'_1 **(resp. k'_2) is the cluster in partition** $\Gamma + 1$ **containing the cluster** k_1 **(resp. k_2) in partition** Γ
 - 12: Update $\Gamma \leftarrow \Gamma + 1$
 - 13: **end while**
-

Discussion. A property of the cannot-link constraint is that when it includes all unique combinations between characters (i.e. every character coexists with all characters in at least one frame), it offers a natural lower bound on the resulting number of clusters, i.e. number of characters. If all characters appear in the same frame (e.g. end of theatrical presentations), the exact number of clusters can be derived by this frame. Typically this is not realistic; in practice, however, principal and secondary characters do share scenes, and therefore the cannot-link constraints aid the clustering. Moreover, by imposing both constraints, C1C reduces the search space of possible groupings, thus resulting in lower computational cost. We also note that the constraints can be imposed to most hierarchical bottom-up clustering methods with minor modifications. For instance, one could use HAC with complete linkage without any specific treatment, as defining cannot-links once is sufficient; whereas, using single linkage would require iteratively checking for constraint violations, similar to C1C.

4 Friends Dataset

Here, we describe the dataset collection and annotation procedure we followed for constructing the Friends dataset, which covers 25 episodes from the third season of the TV show ‘Friends’. Each episode is split into shots (8.8k shots in total), and each shot is then annotated separately. Figure 2 depicts some examples of the principal and secondary characters.

Head track annotation. In this work, we go beyond simply tracking faces, and instead we track and annotate entire heads – including the face (if visible) and the back of the head. We proceed in three stages: (1) head detection and tracking; (2) manual annotations to correct and refine the tracks; and (3) track editing based on these annotations to remove all errors.

Despite the abundance of face detection models [6, 21, 56], head detection, is more challenging, and hence less common. Here, we follow the pipeline from [26] and train a

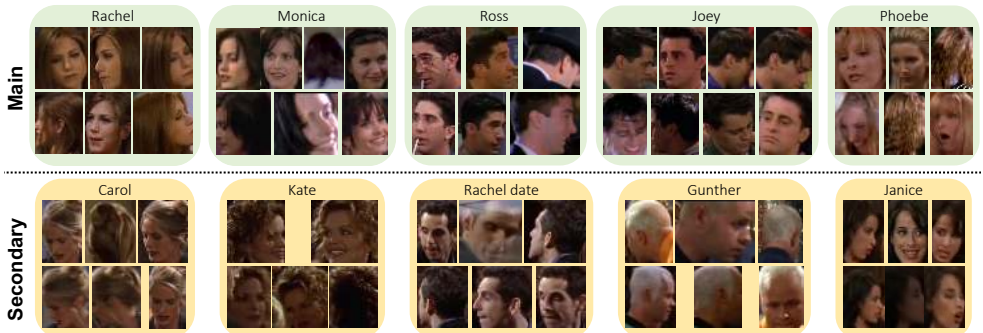


Figure 2: **Friends dataset.** Head crop examples in the Friends dataset for five principal and five secondary characters. Characters are captured in several scenes and viewpoints, including crops showing the back of their heads, thus making the dataset varied and challenging.

head detector [23] on the Hollywood heads dataset [42] to obtain detections for Friends. We construct head tracks by grouping consecutive detections following [20]. We filter out short tracks (less than 10 frames) and tracks with low confidence score. This results in around 33k tracks, i.e. 1.1k tracks per episode. For the track refinement, we manually label tracks as (a) correct (i.e. containing heads), (b) identity switches (i.e. tracks interchanging different characters), and (c) wrong (i.e. not containing heads over time). Then, we keep the correct tracks, and automatically split the ones with identity switches and discard the wrong ones.

Character annotation. To assign character identities to each head track, we first parse character names from IMDB² and keep up to 14 characters per episode, resulting in 49 characters over 25 episodes. Then, we ask annotators to label (using [12, 13]) all head tracks as depicting one of the available characters, as containing irrelevant characters (e.g. a random background person), or as not being valid (e.g. identity switch). This results in about 15k non-valid or irrelevant tracks that we discard following the standard setup [3, 34, 39].

Statistics. Friends contains 17,564 head tracks with approximately 700 tracks per episode. 15,135 tracks depict the six principal characters and the remaining 2,429 depict 43 secondary characters. Compared to other datasets, Friends is $3\times$ larger than the biggest available one (Buffy [41]), and more importantly, it is more varied and contains more challenging cases (see Figure 2) as it contains head track annotations compared to face only tracks [35, 39, 41].

5 Experiments

Here, we describe the datasets and metrics we use (Section 5.1), and we present experimental results, a comparison to the state of the art, and various ablation studies of C1C (Section 5.2).

5.1 Datasets and metrics

In addition to Friends (Section 2), we evaluate C1C on three datasets: BBT [31], Buffy [14], i.e. the two most widely used datasets for video face clustering, and Sherlock [27].

²<https://www.imdb.com/title/tt0108778/>



Figure 3: **Qualitative results of C1C on episode 10 from Friends (principal characters).** Most clusters are pure, i.e. contain only green samples, whereas the mixed ones (last three rows) typically contain challenging samples (red), where it is difficult even for humans to identify the correct character.

The Big Bang Theory (BBT) [31] and Buffy the vampire slayer (Buffy) [14]. They consist of six episodes from the first and fifth seasons of the TV shows and they include manually annotated face tracks. The first releases [31, 39] include tracks with identities for six characters for BBT and twelve for Buffy. As both datasets have been enhanced [34, 39, 41], we use the two most recent releases [34, 41] and group the comparison to the state of the art accordingly. The release from [34] refines some tracks; for this setup, we evaluate on e1 for BBT and e2 for Buffy. The release from [41] contains additional annotations resulting in 103 characters for BBT (five principal, six secondary and 92 background) and 109 for Buffy (six principal, 20 secondary and 83 background). For this setup, we evaluate on all episodes.

Sherlock [27]. It consists of three episodes from the first season of the TV show. It includes manually annotated tracks with identities for 27 different characters (two principal).

Metrics. We report metrics at the frame level, as they compare the quality of C1C when including difficult samples (*e.g.* back of the head) and outliers (*e.g.* not perfect bounding box). To evaluate the cluster quality, we use *Weighted Clustering Purity* (WCP [40, 55]). WCP weights the purity of a cluster by the number of samples belonging in the cluster; to compute it, each cluster is assigned to the class which is most frequent in the cluster. To measure the trade-off between clustering quality and number of clusters, we report *Normalized Mutual Information* (NMI) [25]. Given class and cluster labels Y and C , $NMI(Y, C) = 2 \frac{I(Y; C)}{H(Y) + H(C)}$, where $H(\cdot)$ is the entropy and $I(Y; C) = H(Y) - H(Y \setminus C)$ the mutual information.

5.2 Experimental Results

We validate the effectiveness of C1C by reporting WCP and NMI results on BBT, Buffy, Sherlock, and Friends. Except for some inputs in Table 1 where we also use tracks both

method	train	constraints	source	BBT		Buffy		Sherlock	Friends
				e1	AVG	e2	AVG	AVG	AVG
ULDML [8]	✓	✓	[8]	57.0	–	41.6	–	–	–
HMRf [47]	✓	✓	[14, 30]	59.6	–	50.3	–	–	–
HMRf2 [46]	✓	✓	[30]	66.8	–	–	–	–	–
Imp-Triplet [54]	✓	–	[30]	96.0	–	–	–	–	–
JFAC [55]	✓	–	[14]	–	–	92.1	–	–	–
VDF [33]	✓	–	[3]	89.6	–	87.5	–	–	–
TSiam [34, 35]	✓	–		98.6	–	92.5	–	–	–
SSiam [34, 35]	✓	–		99.0	–	90.9	–	–	–
FINCH [32]	–	–	[34]	99.2	–	92.7	–	–	–
CCL[36]	✓	–		99.6	–	93.8	–	–	–
C1C (ours)	–	✓		99.9	–	94.6	–	–	–
FINCH [32]	–	–		97.0	90.8	77.7	82.9	64.4	69.7
BCL [41]	✓	–	[27, 41]	98.6	93.9	81.4	86.5	–	–
C1C (ours)	–	✓		98.7	95.3	90.1	88.1	76.5	77.0

Table 1: **Comparison to the state of the art on all datasets.** We report %WCP. For BTT and Buffy, we also compare against methods using different (older) versions of the datasets – see text for details.

from [35], for BBT and Buffy we experiment on the tracks provided by BC [41]. We deploy the ResNet-50 network pre-trained on MS-Celeb-1M and fine-tuned on VGGFace2 [5].

C1C performance. The WCP of C1C on BBT, Buffy, Sherlock, and Friends is 95.3%, 88.1%, 76.5%, and 77.0% respectively. Figure 3 illustrates samples from clusters produced by C1C for Friends. We observe that the clusters are quite diverse, as they contain heads from different scenes. The last three rows include failure cases (red). Most of these cases are very challenging, *e.g.* in the cluster Phoebe, the heads of Rachel and Phoebe are confused for one another, or in the cluster Monica the last wrongly assigned head sample is so challenging that even humans looking at it cannot tell that its true identity is Chandler.

C1C vs FINCH. The average WCP of FINCH on BBT, Buffy, Sherlock, and Friends is 90.8%, 82.9%, 64.4%, and 69.7%, respectively. This validates that the must-link and cannot-link constraints improve the performance.

Ablation study. To validate the effectiveness of each component of C1C, we perform an ablation study on BBT e1 and Buffy e2 (Table 2). The first row of Table 2 corresponds to FINCH, the last row to C1C. We observe that adding the must-link constraint brings about +2%; this is expected as instances of the same track are forced together. The cannot-link constraints alone seem very powerful on Buffy and yield a +7.4% boost. Finally, the two constraints are complementary and adding them both yields an even higher performance.

Comparison to the state of the art. Table 1 reports the comparison to the state of the art. We group results that are comparable, *i.e.* ordered by increasing difficulty per dataset release. C1C outperforms all other methods even without requiring any training. C1C outperforms CCL [36] by 0.4% and 0.8% on BBT and Buffy, while increasing the dataset difficulty leads to larger performance gains (*i.e.* +8.69% for Buffy), demonstrating the need for new models.

must	cannot (min-cut)	method	BBT	Buffy
			e2 #C=8	e1 #C=22
-	-	FINCH	97.0	77.7
✓	-	C1C –	98.7	82.0
-	✓	C1C –	97.5	87.3
✓	✓	C1C	98.7	90.1

		Sherlock			AVG S=79
		e1	e2	e3	
	(oracle #C)	27	22	30	
NMI	FINCH [32]	69.3	27.6	47.2	48.0
	C1C	71.5	43.1	45.5	53.3
WCP	FINCH [32]	84.5	41.2	67.4	64.4
	C1C	89.6	66.6	73.2	76.5

Table 2: **Ablation study of C1C on BBT and Buffy.** We report %WCP for FINCH, two stripped-down versions of C1C, and C1C.

Table 3: **Comparison to the state of the art on Sherlock.**

Metric	Method (est #C)	BBT						Buffy						AVG	
		1 7	2 8	3 16	4 18	5 11	6 23	1 17	2 16	3 18	4 22	5 26	6 22		
NMI	FINCH [32]	89.9	88.0	71.2	78.0	75.4	69.5	78.7	74.7	74.4	73.3	72.5	70.4	71.9	72.9
	BCL [41]	95.8	87.2	88.3	76.5	92.2	74.1	85.7	81.7	77.6	77.6	78.1	79.7	78.1	78.8
	C1C	94.1	84.9	90.5	82.2	94.1	80.7	87.8	80.1	85.8	77.0	84.1	83.7	77.4	81.4
WCP	FINCH [32]	96.9	96.4	89.7	91.2	92.8	87.1	92.4	83.9	79.8	77.6	85.6	90.0	83.1	83.3
	BCL [41]	98.6	98.5	90.6	86.9	89.1	81.0	90.8	92.0	79.7	84.0	84.9	89.0	80.5	85.0
	C1C	98.7	97.8	93.3	94.9	94.4	89.4	94.8	90.0	86.4	84.5	88.8	92.4	80.8	87.1
	(oracle #C)	8	6	26	28	25	37		13	22	15	32	38	45	
NMI	FINCH [32]	90.5	83.0	82.5	66.5	88.2	72.1	80.5	78.3	75.0	75.0	77.2	73.8	71.4	75.3
	BCL [41]	92.8	91.9	84.3	78.5	86.1	76.1	84.9	81.3	75.3	77.9	75.9	76.9	78.6	77.6
	C1C	91.0	89.9	83.1	85.1	82.5	75.7	84.5	81.7	86.8	73.4	78.8	74.2	79.5	79.1
WCP	FINCH [32]	97.0	96.4	89.7	86.6	92.4	83.0	90.8	88.9	77.7	80.7	85.6	89.5	74.9	82.9
	BCL [41]	98.6	98.2	92.6	91.7	96.0	86.7	93.9	89.6	81.4	81.4	87.3	91.2	88.5	86.5
	C1C	98.7	97.8	93.9	95.0	95.0	91.3	95.3	88.7	90.1	79.2	89.3	92.4	88.9	88.1

Table 4: **Comparison to state of the art on BBT and Buffy** when using the estimated by BCL [41] number of clusters (est #C) and the ground truth ones (oracle #C). We report the NMI and WCP results for the version of the datasets introduced in [41].

Tables 3-4 report a complementary comparison to the state of the art. We report WCP and NMI for each episode of Sherlock, BBT, and Buffy for the the ground truth number of clusters (oracle #C) and for the number of clusters as predicted by BC (est #C) (for BBT and Buffy only). C1C consistently outperforms FINCH by a significant margin, *e.g.* by 8% for Sherlock, 3-8% for BBT, and 4-8% for Buffy. For BBT, C1C performs on par or outperforms BCL, whereas for Buffy, it outperforms BCL by 2-3%. These results demonstrate the effectiveness of C1C even for challenging cases with secondary and background characters.

Analysis of C1C on Friends. Figure 4 (a) illustrates the WCP over the number of clusters of C1C for the 25 episodes of Friends. The range of accuracy is between 60-90%. We also display the WCP of C1C for the ground truth number of clusters (oracle #C, large solid dots). The final predicted number of clusters (i.e. end of lines) is close to the ground truth one (i.e. large solid dots), suggesting that imposing cannot-link constraints can aid the estimated number of clusters. In Figure 4 (b), we compare the WCP of C1C and FINCH (dotted line) (for visibility, we display six episodes). We observe that the clusters of C1C are more pure than the ones of FINCH and that C1C converges faster, thus demonstrating its superiority.

Furthermore, we examine the correlation between using cannot-link constraints and the performance improvements of C1C over FINCH. In Figure 5, we display the cluster assignments for the principal characters of Friends for (a) FINCH and (b) C1C. Rows correspond to ground truth characters and columns to predictions, *e.g.* ‘Chandler’ is predicted as ‘Joey’

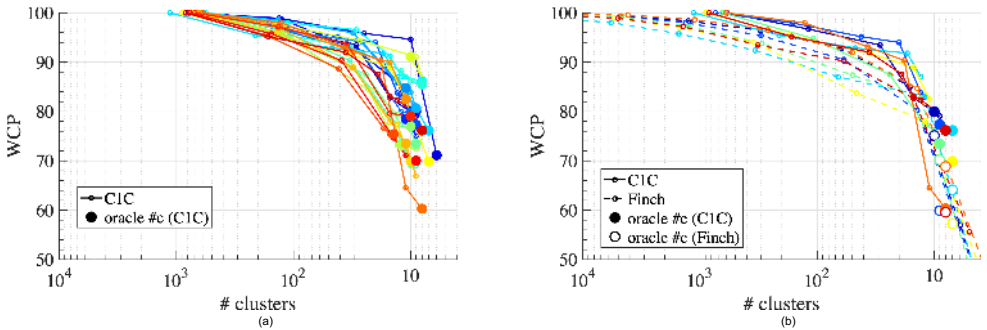


Figure 4: **WCP on Friends.** (a) WCP of C1C on Friends. (b) Comparison of C1C (solid lines) and FINCH (dashed lines) for 6 episodes of Friends. The dots display results for oracle number of clusters: open dots correspond to FINCH and solid dots to C1C. C1C systematically outperforms FINCH.

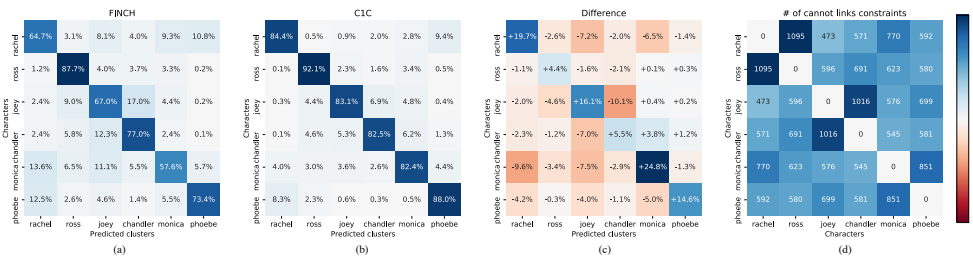


Figure 5: **Correlation between cannot-link constraints and character assignment improvements of C1C over FINCH.** Rows correspond to ground truth characters and columns to predictions. (a)-(b) cluster assignments by FINCH and C1C, (c) their difference, and (d) the cannot-link constraints.

in 12.3% of the cases by FINCH and in 5.3% by C1C. Figure 5 (c) illustrates their absolute difference, *e.g.* ‘Chandler’ is predicted as ‘Joey’ by C1C in 7% fewer cases. Some interesting patterns emerge, *e.g.* the pairs ‘Chandler - Joey’ and ‘Rachel - Monica’ are less frequently mis-assigned. Figure 5 (d) displays the cannot-link constraints. For most character pairs, there is a high correlation between the constraints and the improvement of C1C over FINCH. This validates that the cannot-link constraints result in fewer wrong assignments.

6 Conclusions

We presented C1C, a method for video face clustering. It links instances through first NN relations and imposes must-link and cannot-link constraints acquired in a self-supervised manner. C1C is easily scalable, does not require any training, and requires low computational cost. C1C achieves the new state of the art on several existing datasets for video face clustering, while setting the baseline on Friends, a new dataset we introduce. A future line of work is to automatically estimate the number of characters by using pairwise similarities.

Acknowledgements. We are grateful to D. P. Papadopoulos for helpful discussions, to A. Dutta for the annotation tool, to Q. Pleple and S. A. Koepke for proofreading. Funding was provided by the EPSRC Programme Grant Seebibyte EP/M013774/1.

References

- [1] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *ECCV-Workshop*, 2018.
- [2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *NeurIPS Tutorial*, 2017.
- [3] Martin Bäuml, Makarand Tapaswi, and Rainer Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *CVPR*, 2013.
- [4] Tamara L Berg, Alexander C Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and David A Forsyth. Names and faces in the news. In *CVPR*, 2004.
- [5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2018.
- [6] Dong Chen, Shaoqing Ren, Yichen Wei, Xudong Cao, and Jian Sun. Joint cascade face detection and alignment. In *ECCV*, 2014.
- [7] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*, 2020.
- [8] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Unsupervised metric learning for face identification in tv video. In *ICCV*, 2011.
- [9] Abhijit Das, Antitza Dantcheva, and Francois Bremond. Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach. In *ECCV*, 2018.
- [10] Amit Datta, Anupam Datta, Jael Makagon, Deirdre K Mulligan, and Michael Carl Tschantz. Discrimination in online personalization: A multidisciplinary inquiry. *FAT*, 2018.
- [11] Renato Cordeiro de Amorim. Constrained clustering with minkowski weighted k-means. In *IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, 2012.
- [12] A. Dutta, A. Gupta, and A. Zissermann. VGG image annotator (VIA). <http://www.robots.ox.ac.uk/vgg/software/via/>, 2016. Version: 3.0.7, Accessed: 10 Sep 2019.
- [13] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, 2019.
- [14] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy”—automatic naming of characters in tv video. In *BMVC*, 2006.

- [15] Andrew W. Fitzgibbon and Andrew Zisserman. On affine invariant clustering and automatic cast listing in movies. In *ECCV*, 2002.
- [16] Yue He, Kaidi Cao, Cheng Li, and Chen Change Loy. Merge or not? learning to group faces via imitation learning. In *AAAI*, 2018.
- [17] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018.
- [18] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, 2003.
- [19] SouYoung Jin, Hang Su, Chris Stauffer, and Erik Learned-Miller. End-to-end face detection and cast grouping in movies using erdos-renyi clustering. In *ICCV*, 2017.
- [20] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action Tubelet Detector for Spatio-Temporal Action Localization. In *ICCV*, 2017.
- [21] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *CVPR*, 2015.
- [22] Wei-An Lin, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. Deep density clustering of unconstrained faces. In *CVPR*, 2018.
- [23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.
- [24] Zhengdong Lu. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Artificial Intelligence and Statistics*, 2007.
- [25] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.
- [26] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman. LAEO-Net: revisiting people Looking At Each Other in videos. In *CVPR*, 2019.
- [27] A. Nagrani and A. Zisserman. From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. In *BMVC*, 2017.
- [28] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE PAMI*, 2017.
- [29] Omkar M. Parkhi, Esa Rahtu, Qiong Cao, and Andrew Zisserman. Automated video face labelling for films and tv material. *IEEE PAMI*, 2020.
- [30] Markus Roth, Martin Bäuml, Ram Nevatia, and Rainer Stiefelhagen. Robust multi-pose face tracking by multi-stage tracklet association. In *ICPR*, 2012.
- [31] Anindya Roy, Camille Guinaudeau, Hervé Bredin, and Claude Barras. Tvd: a reproducible and multiply aligned tv series dataset. In *LREC 2014, 9th Language Resources and Evaluation Conference*, 2014.

- [32] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *CVPR*, 2019.
- [33] Vivek Sharma, M Saquib Sarfraz, and Rainer Stiefelhagen. A simple and effective technique for face clustering in tv series. In *CVPR-Workshop*, 2017.
- [34] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Self-supervised learning of face representations for video face clustering. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2019.
- [35] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Video face clustering with self-supervised representation learning. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.
- [36] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Clustering based contrastive learning for improving face representations. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2020.
- [37] Yichun Shi, Charles Otto, and Anil K Jain. Face clustering: representation and pairwise constraints. *IEEE Transactions on Information Forensics and Security*, 2018.
- [38] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna Gummadi, Patrick Loiseau, and Alan Mislove. Potential for discrimination in online targeted advertising. In *ACM FAccT*, 2018.
- [39] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. “knock! knock! who is it?” probabilistic person identification in tv-series. In *CVPR*, 2012.
- [40] Makarand Tapaswi, Omkar M Parkhi, Esa Rahtu, Eric Sommerlade, Rainer Stiefelhagen, and Andrew Zisserman. Total cluster: A person agnostic clustering method for broadcast videos. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, 2014.
- [41] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *ICCV*, 2019.
- [42] VGG. Hollywood heads dataset, 2011. URL <http://www.robots.ox.ac.uk/~vgg/software/headmview/>.
- [43] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, 2001.
- [44] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *ECCV*, 2018.
- [45] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *CVPR*, 2019.
- [46] Baoyuan Wu, Siwei Lyu, Bao-Gang Hu, and Qiang Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *ICCV*, 2013.
- [47] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. Constrained clustering and its application to face clustering in videos. In *CVPR*, 2013.

- [48] Shijie Xiao, Mingkui Tan, and Dong Xu. Weighted block-sparse low rank representation for face clustering in videos. In *ECCV*, 2014.
- [49] Rong Yan, Jian Zhang, Jie Yang, and Alexander G Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *IEEE PAMI*, 2006.
- [50] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, 2019.
- [51] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *CVPR*, 2020.
- [52] Xun Yu, Zicheng Pan, and Yongsheng Gao. A graph-based image clustering method using mutual information maximization. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019.
- [53] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *ECCV*, 2018.
- [54] Shun Zhang, Yihong Gong, and Jinjun Wang. Deep metric learning with improved triplet loss for face clustering in videos. In *Pacific Rim Conference on Multimedia*. Springer, 2016.
- [55] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Joint face representation adaptation and clustering in videos. In *ECCV*, 2016.
- [56] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.
- [57] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015.