Face, Body, Voice: Video Person-Clustering with Multiple Modalities

Andrew Brown[1], Vicky Kalogeiton[1,2], and Andrew Zisserman[1]
[1]VGG, Dept. of Engineering Science, University of Oxford. [2]LIX, École Polytechnique, CNRS, IP Paris
{abrown, az}@robots.ox.ac.uk, vicky.kalogeiton@lix.polytechnique.fr
https://www.robots.ox.ac.uk/~vgg/data/Video_Person_Clustering/

Abstract

The objective of this work is person-clustering in videos – grouping characters according to their identity. Previous methods focus on the narrower task of face-clustering, and for the most part ignore other cues such as the person's voice, their overall appearance (hair, clothes, posture), and the editing structure of the videos. Similarly, most current datasets evaluate only the task of face-clustering, rather than person-clustering. This limits their applicability to downstream applications such as story understanding which require person-level, rather than only face-level, reasoning.

In this paper we make contributions to address both these deficiencies: first, we introduce a Multi-Modal High-Precision Clustering algorithm *for person-clustering in videos using cues from several modalities (face, body, and voice). Second, we introduce a* Video Person-Clustering dataset*, for evaluating multi-modal person-clustering. It contains body-tracks for each annotated character, face-tracks when visible, and voice-tracks when speaking, with their associated features. The dataset is by far the largest of its kind, and covers films and TV-shows representing a wide range of demographics. Finally, we show the effectiveness of using multiple modalities for person-clustering, explore the use of this new broad task for story understanding through character co-occurrences, and achieve a new state of the art on all available datasets for face and person-clustering.*

1. Introduction

Clustering people by identity in videos is an appealing and much-visited topic in computer vision [11, 18, 30, 32, 63, 64, 70]. It has several real-world applications, such as enabling person-specific browsing, organisation of video collections, character based fast-forwards, automatic cast listing; and story understanding, all without requiring any explicit identity labeling. A successful person-clustering framework can therefore alleviate the tremendous annotation cost that is otherwise necessary for such applications.

However, methods for clustering by identity are almost always limited to only using information from faces. Such methods have two significant drawbacks: First, they ignore many available, informative cues that a human would use to solve the task: (i) the person's voice available from the audio track; (ii) the person's overall appearance (from their hair, clothes, posture); and, (iii) the editing structure (in edited material) – such as the co-occurrence of characters in nearby shots and within a scene. Second, they limit the utility of clustering for downstream applications such as story understanding. Understanding the story-line in a scene requires knowledge of *all* the characters present in a scene, not just those whose faces are visible, *i.e. person-level* not face-level reasoning. This is illustrated in Figure 1.



Figure 1: **Video Person-Clustering – an essential step towards story understanding.** Imagine trying to understand the story in the scenes above, given only the *non-greyed-out* parts. Face-level understanding (left) omits important information, such as characters with their backs turned. This work addresses the new task of video person-clustering, which develops *person-level* understanding (right) in a scene by clustering all people, regardless of if their faces are showing or not. This is in contrast to the more limited, established, task of face-clustering. Person-level understanding is essential for downstream applications of grouping-by-identity such as story understanding, and cannot be achieved by face-clustering alone.

Our objective in this paper is to cluster people (or more precisely person-tracks, which depict an entire body in any pose) by identity in movies and TV-material, as a first step towards *story-level understanding*. We cluster people, rather than just faces, and use all cues (face, voice, body appear-

ance, editing structure), including tracks of people from behind without a visible face.

To see the value and necessity of this multi-modal approach, consider the problem of determining if two poor resolution faces depict the same person or not – the voice can discriminatively resolve this ambiguity. Similarly, consider the problem of determining if a person seen speaking to camera in one shot, is the same as the person seen from behind in a following shot – the hair and clothes can provide the link. In Figure 1, for example, how would the people seen from behind be identified other than by clustering their hair, clothes or voice with instances in neighboring shots?

More generally, modalities arising from the same person are both *redundant* and *complementary*, and can be used to address two fundamental problems in clustering: how to obtain *pure* clusters (*i.e*. containing tracks from a single person); and, how to *merge* clusters without violating their purity (*i.e*. by contaminating them with tracks from another person). They can be used to obtain very pure clusters by requiring agreement (*e.g*. on both face and voice) in order for tracks to be grouped together; and can be used to merge clusters which could not otherwise be confidently merged with a single modality, *e.g*. by using the common voice to merge a frontal with a profile face cluster (where the face descriptors of each cluster may be different). In this way, multiple modalities provide a *bridge* between otherwise unmergeable clusters. Methods that merge clusters using a single modality inevitably sacrifice purity.

In this paper, we introduce a new method for the task of video person-clustering, *Multi-Modal High-Precision Clustering* (*MuHPC*), that uses multiple modalities – face, voice, and body appearance. It builds on recent methods that use first nearest neighbour [29, 32, 54] clustering algorithms, and is designed to take advantage of the redundancy and complementarity of the modalities, as discussed above, and to incorporate lessons from the face-clustering literature, such as cannot-link constraints and using the video editing structure [3, 11] (Section 3).

To evaluate the multi-modal person-clustering task, we require a dataset with person-level annotations. However, there are very few such datasets due to the previous emphasis on face-clustering and moreover, most face-clustering and labelling datasets, such as Buffy [16] and TBBT [53], are based on TV material with limited diversity in skin color. For these reasons, we introduce a new *Video Person-Clustering Dataset* (*VPCD*) where we: (i) re-purpose multiple existing face datasets by adding person-level multi-modal annotations (*e.g*. all person-tracks and voice utterances); and (ii) include different TV shows and films (hereby referred to under the unified term *program sets*) to address this lack of diversity. *VPCD* consists of visually disparate program sets, and includes body-tracks, face-tracks when visible; and voice utterances when speaking, for all annotated characters. We provide features so that future clustering algorithms can be compared easily and fairly (Section 4).

We show the effectiveness of multi-modality and outperform strong baselines for person-clustering on *VPCD* (Section 5.1), and explore this new expansive task for story understanding (Section 5.4). Our method also significantly outperforms the face-clustering state of the art on both TBBT and Buffy by over 10% NMI (Section 5.3). Note that our goal is multi-modal clustering and not representation learning. Thus, we do not propose a new architecture or train a network for better features. Instead, we use features from pre-trained networks (for face and speaker recognition) and only train a network where it is necessary for body Re-ID. A broader impact statement is included in the appendix.

2. Related Work

In this work, we focus on multi-modal person-clustering in videos. Similar works target the more limited task of face-clustering or labelling, person Re-ID, or person search. We describe them, and also discuss similar datasets to *VPCD*.

Face-Clustering. A well-studied task for both images [4, 24, 44, 52] and videos [32, 63, 64, 75], with difficulty arising from the variation of pose, lighting, and emotion [23, 36] in faces of the same identity. Most video approaches exploit the spatio-temporal continuity and find must-link and cannot-link clustering constraints [3, 11, 14, 32, 55, 57, 64, 66, 70, 72], or additional constraints from the structure of videos [64]. Most works approach face-clustering with metric or representation learning [11, 18, 55, 56, 57, 63, 69, 70]. For instance, [63] map features from the same identity to a fixed-radius sphere, while Sharma *et al*. use supervision from video constraints [55, 56] or weak clustering labels [57]. These methods, however, are limited by the relatively small training sets available from particular TV-shows. For this reason, some recent approaches focus on simply clustering pre-trained features that have been learnt on very large-scale face datasets. [54] propose a simple first nearest neighbour clustering method upon pre-trained features, FINCH, and show impressive results. More recently, [32] combines [54] with spatio-temporal constraints and improves performance. *All* the above works focus on the limited task of clustering faces (Figure 1 - left), whereas our focus is multi-modal person-clustering *i.e*. clustering every appearance of characters, regardless of whether their face is visible (Figure 1 - right), and using multiple modalities.

Face-Labelling. The task of classifying faces by identity - most works address this by using face-appearance with supervision from transcripts aligned to subtitles [3, 5, 13, 16, 17, 46, 49, 58, 61], for example by using Multiple Instance Learning [5, 21, 33, 68]. Some exploit cues other than faces from videos: [48] use clothing to match faces in TV-shows across shot boundaries, while [6, 43] use face and voice to label faces. [47] use face and voice to retrieve a list of

shots containing a named person, by searching for their name in subtitles and displayed text. These works focus only on visible faces and although some are multi-modal (face, voice and/or text supervision), the text is typically obtained from external sources (*i.e.* transcripts). Our task is different, as we cluster rather than label, and thus do not require character-classifiers or ID supervision or extra annotation, and we use all available cues *i.e.* editing structure and multi-modality.

Person Re-ID. The task of re-identifying pedestrians in CCTV - typically [35, 67, 76, 77], each body is fully visible and walking, the clothing remains constant for each identity, and the images are low resolution. This differs substantially from person-clustering in TV and film material, where there is large pose variation (*e.g.* sitting, standing, lying down), occlusion, and the clothing frequently changes for each identity. A full literature review is out of scope. Closer to our task are works on person-retrieval in photo albums [31, 59, 74] or person-search from portraits in videos [27, 71]. [27, 59] use face and body features, while [71] use audio. The TRECVID Instance Search challenges [1] involved retrieving a list of shots that contain an identity, given a query video for that identify. In contrast, we cluster all characters at the track-level in videos without requiring search queries.

Related Datasets. Various face-clustering datasets have been proposed [12, 16, 19, 32, 45, 53]. These follow some similar trends: (a) are limited in size, consisting of a movie or some TV show episodes; (b) under-represent most demographic groups; and (c) contain only face annotations, so cannot be used for the broader multi-modal person-clustering task. Several story understanding [2, 28] or person-search [27] datasets with face and/or body annotations exist. These cannot be used for our task, as they lack audio [27, 28] or contain only partial annotations such as keyframes [28] or for a subset of tracks [2]. Furthermore none contain labelled voice utterances. Instead, *VPCD* contains 6 different TV-shows and movies, representing a more diverse range of characters, and containing *multi-modal annotations* for all annotated characters.

Story Understanding. This targets automatic understanding of human-centred story-lines in videos. It has been formulated in several ways, *e.g.* grouping scenes by story threads [15, 50], learning character interactions [39, 62] or relationships [34], creating movie graphs [65]; or text-to-video retrieval from narrating captions [2], with several datasets [2, 28] introduced. Many works [2, 34, 65] highlight the importance of knowing who is present in a scene for understanding the story. This is the focus of our work.

3. Method

Here, we describe the *Multi-Modal High-Precision Clustering* (*MuHPC*) method for person-clustering in videos. It is a single hierarchical agglomerative clustering [51] (HAC) approach that groups person-tracks by identity using similarities of modality features, together with constraints arising from the video structure. *MuHPC* uses pre-computed features, and hence does not require any training outside of simply learning optimal hyper-parameters, and can then run out of the box for any video dataset. In this work, we use three modalities (face, voice, and body appearance) but *MuHPC* can easily scale to any number of modalities.

Overview. *MuHPC* consists of three stages (Figure 2). **Stage 1** creates high-precision clusters using a single modality, here face. We group person-tracks that share a first nearest neighbour (NN) using multiple iterations of HAC, as in [29, 32, 54]. We follow this trend subject to two additional *constraints*: a cannot-link constraint for concurrent tracks (as in [32] based on [3, 11]), and a conservative threshold on the maximum NN distance. This results in K_1 clusters (Section 3.1). **Stage 2** exploits multi-modality to *bridge* clusters that were otherwise unmergeable by the single face modality with a conservative threshold; in particular, by requiring that different modalities (*i.e.* face and voice) concur on the merge (Section 3.2). **Stage 3** clusters tracks without visible faces, and hence that are not yet clustered by the first two stages. Constraints from the editing structure (neighboring shots) and a conservative threshold on body features (so that they depict the same person with the same clothing) are used to link face-less person-tracks to clusters with faces (Section 3.3). Here, we describe the stages, algorithm design choices, and how the hyper-parameters are learnt.

Notation. Given a dataset with person-tracks and C characters, where x_i is a single person-track, the goal is to cluster all x_i by identity into C clusters (C is unknown). Each person-track x_i is represented by one feature vector per available modality, *i.e.* $x=\{x_f, x_v, x_b\}$, with x_f, x_v, x_b the face, voice and body-track features, respectively. The availability of each feature vector is dependant upon the part of the person that is visible (face and/or body), and if they are speaking. For each person, at least one of x_f, x_b are available. Let $d(x_i, x_j)$ be the distance between two track features of the same modality, and d_f, d_v and d_b the distances between two face, voice or body-tracks, respectively; the lower the value, the more likely the tracks depict the same identity. NN is nearest neighbor; $n^1_{x_i}$ is the first NN track of track x_i. The set of video frames that x_i is present in is denoted by T_i.

3.1. Stage 1: High-Precision Clustering

Stage 1 creates high-precision clusters, each containing tracks of the same identity. It uses only the face modality as this is the most discriminant of the three (face, voice and body), and thus is least likely to group different identities in the same cluster. Here, we use a NN clustering method [32, 54], subject to two clustering constraints.

Clustering Constraints. A NN is only considered valid if the resulting merge satisfies: *(1) A Spatio-Temporal Cannot-link Constraint:* Tracks that have (partial) temporal overlap

cannot be grouped together, since they must represent different characters as they appear together in at least in one frame (introduced by [32]); and *(2) A NN Distance Constraint:* the distance $d_f(x_i, n^1_{x_i})$ between a track x_i and its first NN $n^1_{x_i}$ is less than a strict threshold τ_f^{tight} for Stage 1.

Clustering process. At every iteration (cluster partition Γ), each cluster is grouped with its NN cluster, *i.e.* the closest. Specifically, the first partition groups tracks into clusters through first NN relations, while following partitions group the clusters formed in the previous partition; each cluster is represented by the average of the features it contains. Following the notation of [54], at each partition Γ, the method forms K_Γ clusters by merging tracks that are either first NN (mutually or one is the first NN of the other) or have a common NN $n^1_{x_i}$, as described by the adjacency matrix:

$$A(x_i, x_j) = \begin{cases} 1 & \text{if } \left(x_j = n^1_{x_i} \text{ or } n^1_{x_j} = x_i \text{ or } n^1_{x_i} = n^1_{x_j}\right) \\ & \text{and } T_i \cap T_j = \emptyset, d_f(x_i, n^1_{x_i}) \leq \tau_f^{\text{tight}} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Discussion. In standard HAC the clustering continues until all clusters merge to one. Including the constraints introduces strict stopping criteria, and therefore the clustering stops when either the clusters are all more than a distance τ_f^{tight} apart, or they are separated by a cannot-link constraint. This results in K_1 high-precision clusters, where we expect $K_1 \geq C$. The very simple addition of a distance threshold leads to a significant improvement in clustering results over [32, 54, 63] (Section 5.3). Without this constraint, little prevents an incorrect merging of clusters of different identities and the subsequent creation of low-precision clusters.

3.2. Stage 2: Multi-modal Cluster Bridging

Combining a discriminative modality with the constraints results in *high-precision* clusters. However, a single modality alone cannot continue making confident merges without sacrificing purity. Thus, Stage 2 merges these clusters by exploiting multiple modalities *i.e.* face and voice.

Modality-pair merges. To further merge clusters, we demand that two modalities agree that the clusters contain the same identity. Therefore, we require that the distances for the face and voice are both below new thresholds, *i.e.* $d_f < \tau_f^{\text{loose}}$ and $d_v < \tau_v^{\text{loose}}$. Note, here we use features taken from tracks within clusters, rather than averaged cluster features. τ_f^{tight} is raised by just a small margin, δ, *i.e.* $\tau_f^{\text{loose}} = \tau_f^{\text{tight}} + \delta$, due to the concurrent agreement from the voice.

Discussion. This stage results in K_2 clusters with high-precision, where $K_2 \leq K_1$. Here, we use face and voice as they have been shown to be coupled [41, 42] and to contain redundant, identity discriminating information. An alternative is to require that the voice modality alone provides a confident (*i.e.* tight threshold) match, *e.g.* two person-tracks with the same voice. We find however that voice alone cannot reliably join clusters of the same identity. This can be because two identities with the same emotion in their voice (*e.g.* shouting, crying) can appear similar to the less discriminative voice embedding (more in the appendix).

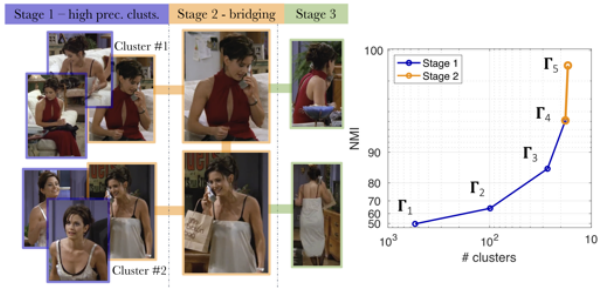


Figure 2: **The clustering process of *MuHPC*.** (Left) Example person-tracks at each stage of *MuHPC*. Two high-precision clusters from Stage 1 depicting the same character. One contains near-frontal faces (below) and one profiles (top), hence the single face modality cannot confidently merge the two. Stage 2 uses a talking person-track from each cluster to form a bridge, by demanding the agreement of both face and voice modalities that these contain the same identity. Stage 3 merges face-less bodies into the formed cluster. (right) The NMI and number of clusters at each partition, Γ, of stages 1 and 2 on an example video from *VPCD*. At each partition the number of clusters decreases, while the normalised mutual information increases. At Γ_4 Stage 1 clustering stops. Stage 2 progresses to Γ_5 by bridging clusters. Stage 3 does not affect the number of clusters.

3.3. Stage 3: Clustering backs

Stages 1 and 2 result in high-precision clusters. Nevertheless, they do not account for person-tracks with no visible face, for instance when viewed from behind, *i.e.* a *face-less* person. The goal of Stage 3 is to add the face-less person-tracks into their respective high-precision clusters using the modality of body-appearance. Here, we use the editing structure of the videos, given that the appearance of the same character can change dramatically between scenes. As discussed above, body features may not be discriminative for identifying if characters are wearing very similar clothing. We determine such body-tracks using the simple ratio-test introduced in [37]. Specifically, for each body-track we compute the first and second NN distances, d^1_{b,x_i} and d^2_{b,x_i}. If the ratio, $d^1_{b,x_i}/d^2_{b,x_i}$ is higher than a threshold ρ then the body-track is classified as non-distinctive and is ignored.

For assigning face-less people to clusters, we find the NN body-track (that has a face and hence is already clustered) that does not violate the ratio-test in a neighbouring shot, and assign the face-less person to this cluster. Given that the same person is most likely wearing the same outfit in the same or neighbouring shots, we only examine the distance between body-tracks from these shots. At this stage, some backs cannot be clustered with high confidence, either because they are not similar to any nearby body or because

they fail the ratio test for being a non-distinctive feature. Our design choice is to ignore these backs, *i.e.*, we ignore any back for which the NN distance is more than a threshold τ_b^{back}. Note, this stage keeps the number of clusters to K_2.

Required Number of Clusters. Suppose we know the number of characters C, and hence the number of clusters. Our goal is to reduce K_2 to the desired C (typically $K_2 \geq C$). Previous methods [63] employ HAC; however, this suffers from reliance on features that can no longer confidently discriminate between clusters of the same person. Instead, we employ a cluster prior: there is no identity overlap amongst the largest clusters *i.e.* they contain unique identities, and conversely there is likely an identity overlap between a small and large cluster. Our intuition is that big clusters contain ample information about an identity, and consequently if two large clusters contained the same identity, then they would have been merged. Therefore, we iteratively merge the smallest with the largest cluster until there are C clusters. In practice, we observe that small clusters contain blurry or low-resolution tracks, and so could not confidently be merged at earlier stages.

Discussion. Most methods [55, 57, 63] fine-tune character features on a video dataset. Instead, *MuHPC* operates on pre-trained features, thus reducing the computational burden and leading to increased generalisation capabilities. An extension would be to replace the constraints with a cost function optimisation approach, allowing a cannot-link to be correctly broken for a person's reflection in a mirror.

3.4. Learning Hyper-Parameters

The hyper-parameters for *MuHPC* are learnt on the validation partition of *VPCD*. The visually disparate program sets in the test partition are disjoint from those in the validation, yet these parameters are kept constant. For the hyper-parameter associated with the face modality (τ_f^{loose}) this is possible as the face features are trained on millions of faces [8], and therefore are highly discriminative and universal (generalise well across different program sets). However, voice identity features are less universal than face features, and hence there is not a single good choice for τ_v^{loose} that would generalise across the audibly disparate program sets. Instead, we learn a unique value *automatically* for each. Our goal is to choose τ_v^{loose} to be lower than the minimum distance between voices from different people. The cannot-link constraints automatically provide face-track pairs of different identities. We measure the distances between different people's voices. In practice, there are too few constraints between speaking faces to provide an accurate representation of the negative distances, as rarely two face-tracks speak in the same shot. We combine the cannot-link speaking face-tracks with clusters from Stage 1 to provide more examples. This leads to many negative distances and an accurate representation of their distribution. We select τ_v^{loose} as the lower 99.9 percentile of these distances. This provides a robust automatic threshold measure. For program sets with similar sounding characters, this process gives a low τ_v^{loose} (*e.g.* Buffy – many similar sounding teenagers).

Dataset	#eps	length	#IDs	Gender F/M	#Tracks body	face	voice
TBBT [53]	6	2h 6m	103	53/50	4,276	3,908	1,047
Buffy [16]	6	4h 9m	109	37/70	7,561	5,832	1,835
Sherlock [43]	3	4h 30m	31	16/15	6,232	6,247	1,615
Friends [32]	25	9h 22m	49	23/26	18,360	17,333	3,961
ALN [60]	1	1h 40m	10	4/6	1,932	1,614	404
HF [60]	1	2h 7m	24	11/13	1,416	1,463	303
VPCD		23h 54m	326		39,777	35,396	9,165

Table 1: ***Video Person-Clustering Dataset* statistics.** For each program set in *VPCD* we detail video and annotation statistics. #eps: number of episodes; #IDs: number of unique characters; TBBT: The Big Bang Theory; (movies) ALN: About Last Night; HF: Hidden Figures. We cite the first published work that used each respective program set for face-clustering, but we provide additional full multi-modal annotations for each.

4. Video Person-Clustering Dataset

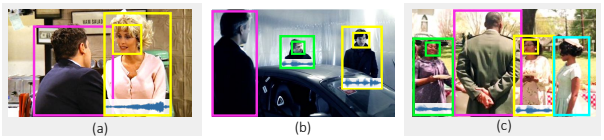


Figure 3: ***VPCD* dataset.** It consists of different and diverse TV shows and movies; here, we display a subset of them: (a) Friends, (b) Sherlock, (c) Hidden Figures. *VPCD* contains face, body and voice tracks annotated for many characters. Here, we display such examples. Each face-body pair is displayed with a unique color. A more representative range of characters are captured in a variety of scenes (*e.g.* dark (b)), viewpoints (*e.g.* (c)); and poses, including backs of bodies (magenta, cyan). When speaking, we also include a voice-track (blue signal below body-tracks).

In this section, we describe the dataset (Section 4.1), the annotation (Section 4.2), and the feature extraction processes (Section 4.3). The dataset is built on top of existing video datasets that have face-level annotations (labeled face-tracks) by adding and annotating body-tracks, and annotating voice utterances. This is for three reasons: first, it enriches the existing dataset by raising them to have person-level annotations; second, it enables comparisons on face-level clustering with prior work on these datasets; and third, it means that the video material is already publicly available and we need only release the new annotations (and features).

4.1. *VPCD* content

VPCD contains *full multi-modal annotations* for primary and secondary characters for a range of diverse and visually disparate TV-shows and movies (statistics in Table 1, examples in Figure 3). *VPCD* contains annotations for 39,777 body-tracks, 35,396 face-tracks for whenever the face is visible, and 9,165 manually annotated voice-tracks for when-

ever each of them are speaking. Identity discriminating features (embeddings from deep networks) are provided for all modalities. A total of 23 hours of video cover a range of genres and styles such as Hollywood Drama (Hidden Figures, 2016), Romance (About Last Night, 2014), fast-paced Action/Mystery (Sherlock, Buffy) and live studio-audience sitcoms (Friends, TBBT). A large variety of characters are annotated, ranging from small casts shown over many episodes (*e.g.* Friends) to program sets with a long-tailed distribution with many secondary/background characters (*e.g.* Buffy). *VPCD* is by far the largest dataset of its kind. The program sets were chosen such that *VPCD* is representative of the diversity of people's appearance in the real world. There is a validation set and a test set - these are disjoint. The validation set is the first five episodes of Friends.

4.2. Annotation Process

Here, we describe the annotation process for the face, body, and voice tracks in *VPCD*. For all component program sets, the face annotations already exist, and define the characters of interest for that video. Our goal is to annotate their body and voice-tracks. Very often in videos, a character is seen facing from behind (Figure 3). This means that the existing face-tracks cannot be used to trivially annotate the body-tracks by spatial overlap (since there will be no face-track). We therefore combine automatic and manual annotation methods (more details in the appendix).

Face. We use the same face bounding-box/track annotations and ID labels as were provided with the original datasets so that we can compare to previous works on face-clustering.

Body. We detect bodies with a Cascade R-CNN [7] trained on MovieNet [28] and form tracks with an IOU tracker. When a body-track clearly corresponds to a face-track (*i.e.* no significant IOU with any other face-track), the body-track is automatically annotated with the character name of that face-track. We manually annotate the remainder as well as the body-tracks corresponding to characters from behind.

Voice. We manually segment the audio-track into the speaking parts for all annotated characters. To ensure the correctness of the segmentation, the audio track was first segmented by one human annotator, and then verified by different ones.

4.3. Feature Extraction

Face. We use L2-normalised 256D features, extracted from an SENet-50 [26] pre-trained on MS-Celeb-1M [20], and fine-tuned on VGGFace2 [8] (same as [16, 32, 43, 53]).

Body. For all body detections, we extract 256D features with ResNet50 [22] trained on CSM [27]. We average the features across each body-track, and then L2-normalise them.

Voice. Following [9], we extract a single, L2-normalised 512D speaker embedding from each voice segment using a thin-ResNet-34 [22, 73] trained on VoxCeleb2 [10].

5. Experiments

Here, we evaluate *MuHPC*. We first give experimental details, followed by person-clustering results on *VPCD* and provide ablations. We compare to previous face-clustering works and finally examine the advantages of person-clustering for story understanding. Further ablations and experiments on clustering all characters in all videos simultaneously are included in the appendix.

Implementation details. We use the face, body and voice track annotations and features from *VPCD* (Sections 4.1,4.3). For all modalities, feature distances d_f ,d_b,d_v are computed using (1 - cosine similarity). As described in Section 3.4, parameters are learnt on the *VPCD* val. set. The values are: τ_f^{tight}=0.48, δ=0.025, ρ=0.9 and τ_b^{back}=0.4. These parameters are fixed for all experiments, and only have to be re-learnt if the features change. Details on the automatically selected τ_v^{tight} values are in the appendix.

Metrics. For each dataset in *VPCD*, we measure each metric at the episode level and average over all episodes. Following [32, 63], we use Weighted Cluster Purity (**WCP**) and Normalized Mutual Information (**NMI**). WCP weights the purity of a cluster by the number of tracks belonging in it. NMI [38] measures the trade-off between clustering quality and number of resulting clusters. **Character Precision and Recall (CP, CR)** are computed using the number of ground truth identities. Each identity is uniquely assigned to a cluster. CP is the proportion of tracks in a cluster that belong to its assigned character, while CR is the proportion of that character's total tracks that appear in the cluster. They are averaged across all characters, thus weighting each equally.

Test protocol. We evaluate: (i) automatic termination (AT), *i.e.* unknown number of clusters, and (ii) oracle cluster (OC), when known. AT is realistic for applications, while OC offers a fair comparison to the state of the art.

5.1. Person-Clustering

Baselines. To evaluate person-clustering, we compare to two strong baselines stemming from the best existing face-clustering algorithm, C1C [32]. The first, B-ReID, is inspired by person Re-ID [35, 76, 77] and uses C1C to cluster body rather than face features. It ignores person-tracks without bodies (<2% of person-tracks). For the second, B-C1C, we use regular C1C to cluster faces, with the addition of Stage 3 of *MuHPC* for clustering face-less bodies.

Results and analysis. Table 2 reports person-clustering results when testing on *VPCD*. For all metrics, *MuHPC* (full method) significantly outperforms the strongest baseline by on average 6.1% in WCP and 11.8% in NMI. B-ReID is poor due to frequent clothing changes. *MuHPC* outperforms B-C1C thanks to (1) the NN distance threshold that prevents incorrect merges and subsequent low-precision clusters, and (2) the multi-modal bridges that merge clusters which face

#	Modality			TBBT				$\#C_s$= 130	Buffy				$\#C_s$= 165	Sherlock				$\#C_s$= 50	Friends				$\#C_s$= 239	Hidden Figures				$\#C_s$= 10	About Last Night				$\#C_s$= 24	Average				$\#C_s$= 618
	F	B	V	WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR	
B-ReID		✓		80.5	69.7	49.6	55.0		65.0	60.9	52.7	46.8		61.2	28.9	43.6	44.3		70.9	60.4	71.0	56.3		32.6	23.4	36.8	19.6		41.0	14.1	37.4	32.6		58.5	42.9	48.5	42.4	
B-C1C		✓	✓	87.7	69.2	39.4	50.6		73.6	58.2	34.6	41.6		77.7	41.6	29.3	43.6		85.3	77.1	69.5	70.8		76.2	69.8	55.2	50.3		94.4	85.8	68.0	76.8		82.5	67.0	49.3	55.6	
MuHPC–	✓			93.5	84.6	76.4	77.6		80.0	66.7	63.8	65.2		83.8	52.3	51.2	**58.4**		85.7	73.7	81.3	79.0		**77.6**	**70.4**	**59.1**	**52.1**		95.7	89.7	98.2	86.3		86.1	72.9	71.7	69.8	
MuHPC$_v$	✓		✓	93.5	84.6	76.4	77.6		80.1	67.2	64.2	64.7		84.5	59.3	54.9	57.3		86.9	75.3	84.0	82.8		77.6	70.4	59.1	52.1		96.0	**90.5**	98.3	86.4		86.4	73.5	72.3	69.7	
MuHPC$_b$	✓	✓		**96.9**	**92.8**	**80.4**	**79.6**		85.7	75.6	68.1	**67.9**		84.1	52.9	51.7	54.3		89.5	81.3	84.6	82.4		77.6	70.3	59.0	52.0		95.7	89.4	98.2	86.3		88.2	77.1	74.0	70.0	
MuHPC	✓	✓	✓	96.9	92.8	80.4	79.6		**85.8**	**76.4**	**68.4**	67.2		**84.8**	**60.0**	**55.2**	57.2		**90.8**	**83.1**	**87.7**	**86.6**		77.6	70.3	59.0	52.0		**96.0**	90.2	**98.3**	**86.4**		**88.6**	**78.8**	**74.8**	**71.5**	

Table 2: **Person-Clustering Results on *VPCD*.** For each program set, each metric is averaged across all episodes. AT protocol. The 'Average' column reports averaged metrics across all six program sets. $\#C_s$ is the sum of ground truth clusters across each episode in each program set. We report two strong baselines (B-ReID, B-C1C, Section 5.1) and an ablation on the modalities used. Keys: F-face, B-body, V-voice. *Modality*: used modalities.

alone cannot. This validates that using all available video cues, such as multi-modality and editing structure aids video person-clustering substantially. The clustering process for a character in *VPCD* is visualised qualitatively and quantitatively in Figure 2. *MuHPC* improves most upon the baselines on the more unconstrained program sets with many secondary characters and long-tailed character distributions (*e.g.* TBBT, Buffy, Friends, Sherlock). Here, *MuHPC* uses the NN distance threshold to keep the clusters of the many characters separated, and then merges any repeated clusters of main-characters via talking person-tracks. The *MuHPC* clustering process is visualised in Figure 2.

5.2. Ablation

Here, we perform ablations on the different modalities in *MuHPC*. Detailed results and parameter sweeps can be found in the appendix. Table 2 includes an ablation of the multi-modality, *i.e.* using voice (*MuHPC*$_v$ – Stage 2) or body (*MuHPC*$_b$ – Stage 3) modalities or both (*MuHPC*). Experiments without the body modality do not use Stage 3, and instead cluster each face-less body to the temporally-closest (Temporal-NN) body with a face in a nearby shot. Due to the threading structure [25] of edited videos, there is a strong prior that the Temporal-NN is correct.

Adding either the voice or body offers a benefit over *MuHPC*–, due to the increased discriminative capabilities from an additional modality. *MuHPC*$_b$ outperforms *MuHPC*$_v$, as there are many face-less bodies in *VPCD*, and the body modality allows for these to be clustered correctly. Using the voice in conjunction with the body (*MuHPC*) performs best, as their benefits are compounded, and the multi-modal bridges connect clusters with higher purity. The voice gives a higher boost when used alongside the body modality, as otherwise the multi-modal bridges are merging lower precision clusters. The voice adds significant benefit in NMI on multiple program-sets. This is impressive as the tight voice thresholds were found *automatically*. Sometimes the voice does not lead to an improvement, due to the absence of speaking person-tracks in merge-able clusters (*e.g.* TBBT). Additionally, the body offers little improvement in the two movies (Hidden Figures, About Last Night) that have many dark scenes and non-distinctive clothing. Here temporal-NN is able to assign face-less bodies to clusters well. Note, NMI increases more than WCP when adding the voice modality, because bridging two high-precision clusters will not greatly effect the purity; however, it leads to increased NMI as there is less identity overlap between the resulting clusters.

MuHPC requires manually diarised speech segments. Preliminary results show that automatic diarisation methods lead to smaller improvements from the voice modality than when manually diarised voice is used, but we leave this to future work. We highlight that with 24 hours of manually diarised audio, VPCD provides a unique test bed for future research on moving beyond requiring manual diarisation.

5.3. Face-Clustering

Here, we compare to previous works by experimenting only on face-tracks, excluding person-tracks without faces. We compare to FINCH [54] (evaluated at the required number of clusters, from [32]), BCL [63] and C1C [32]. For TBBT and Buffy, the face annotations are the same as [32, 63]. Here, we do not compare to works that use the less challenging [32] subset of the annotations [52, 57]. For our method, we present: (i) *MuHPC*– uses only face-tracks, *i.e.* exactly the *same* information and features as other methods, hence results are directly comparable; and (ii) *MuHPC*$_v$ uses face-tracks with multi-modal bridges (*i.e.* voice). Following [32, 63], performance is evaluated at frame level.

Table 3 reports face-clustering results. For both AT and OC protocols, *MuHPC*– significantly outperforms the state of the art in all metrics, as it avoids incorrect merges, hence maintaining cluster purity. For instance, NMI, CP and CR boost by +10-14% for Buffy and TBBT for OC, and by over 10% for WCP averaged across all datasets for AT. *MuHPC*$_v$ also leads to a boost over *MuHPC*– in most datasets. We observe that the more challenging the dataset, the higher the boosts by multi-modality, *e.g.* +3.8% in CR for Friends and +7.4% in NMI for Sherlock. We note that on NMI, WCP, the performance on TBBT is now almost saturated. A full discussion of results is given in the appendix.

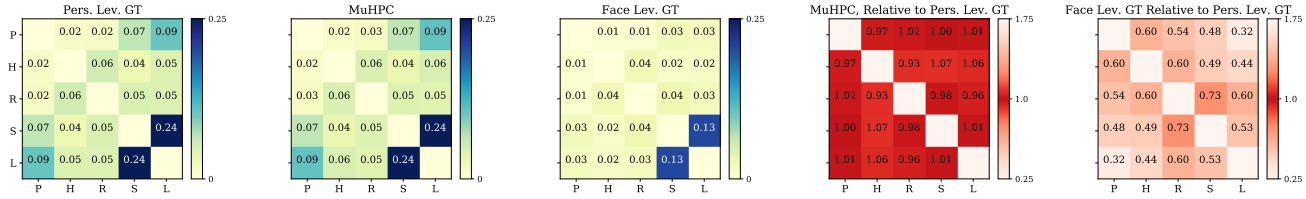


Figure 4: **Character co-occurrences for story understanding** between the 5 main characters in the 6 episodes of The Big Bang Theory in *VPCD*. (a) The ground truth co-occurrences as a proportion of the combined temporal length of the 6 videos. It is generated from *VPCD* which annotates each character whenever they are visible. Higher indicates more co-occurrence. (b,c): *MuHPC* and face-level clustering co-occurrences, as a proportion of the 6 videos; (d,e): *MuHPC* and face-level co-occurrences, relative to the ground truth (a). (d,e) are obtained by dividing (b,c) by (a), respectively. 1.0 indicates that the prediction is the same as the ground truth. Key: P: Penny, H: Howard, R: Raj, S: Sheldon, L:Leonard.

Method	protocol	TBBT				$\#C_s$ = 130	Buffy				$\#C_s$ = 165
		WCP	NMI	CP	CR	$\#C_p$	WCP	NMI	CP	CR	$\#C_p$
BCL [63]	AT	90.8	85.7	-	-	83	85.0	78.8	-	-	121
C1C [32]	AT	89.2	87.4	29.1	40.9	41	66.3	68.8	14.9	27.1	40
MuHPC–	AT	**99.4**	**97.8**	**87.8**	**88.6**	168	**96.1**	92.8	85.6	**85.5**	223
MuHPC$_v$	AT	**99.4**	**97.8**	**87.8**	**88.6**	168	**96.1**	**93.7**	**85.9**	84.8	221
Finch [54]	OC	90.8	80.5	46.1	44.2		82.9	75.3	49.6	41.0	
BCL [63]	OC	94.0	85.0	-	-		86.5	77.6	-	-	
C1C [32]	OC	95.3	84.5	54.9	57.3		88.1	79.1	58.1	55.4	
MuHPC–	OC	**99.1**	**97.4**	**79.3**	**83.0**		**95.6**	92.2	**72.3**	**73.8**	
MuHPC$_v$	OC	**99.1**	**97.4**	**79.3**	**83.0**		**95.6**	**93.1**	71.5	73.2	
		Friends				$\#C_s$ = 239	**Sherlock**				$\#C_s$ = 50
C1C [32]	AT	88.2	89.8	62.4	73.2	185	76.3	50.3	20.2	41.0	25
MuHPC–	AT	**98.7**	94.9	98.1	94.0	543	**86.7**	60.3	79.1	71.2	96
MuHPC$_v$	AT	98.4	**95.9**	97.7	**95.3**	522	86.3	**66.0**	**78.4**	**74.5**	86
Finch [54]	OC	92.2	89.9	85.2	85.6		81.6	58.6	**59.8**	56.8	
C1C [32]	OC	94.3	93.2	79.1	85.5		81.6	53.8	40.5	51.7	
MuHPC–	OC	96.3	92.7	89.0	88.8		84.0	56.5	55.4	59.9	
MuHPC$_v$	OC	**97.1**	**94.6**	**92.3**	**92.6**		**85.1**	**63.9**	59.6	**62.9**	

Table 3: **Face-Clustering Results.** Comparisons to previous state of the art on four program sets using only face-tracks with unknown (AT), and known (OC) number of clusters. We report metrics averaged over each episode in each program set, and the number of predicted clusters, summed over each episode ($\#C_p$). *MuHPC*– uses only face; *MuHPC*$_v$ uses the multi-modal bridges from voice and face. Where not reported in respective publications, numbers are computed using official implementations. Finch has no stopping criterion so results for AT are not reported.

5.4. Enabling Story Understanding

Here, we explore how close we have come to enabling story understanding. Clustering people (rather than faces) indicates who is present in a scene (Figure 1) – an essential and necessary step for predicting character co-occurrences, and hence their interactions [34, 40] that make up a story. Specifically, we ask two questions: (1) Can clustering on the face-level predict the co-occurrence of two characters correctly? (2) How close is *MuHPC* to correctly predicting co-occurrences? To answer these, we experiment with the five main characters from the six episodes of TBBT in *VPCD*. Figure 4a shows the ground truth (Pers. Lev. GT) co-occurrence of character pairs as a proportion of all frames in the show, and hence is a measure of their interaction, *e.g.* Sheldon and Leonard co-occur for 24% of all frames.

For the first question, we visualise the co-occurrences of characters according to the face-track annotations (Face Lev. GT) in *VPCD* (shown in absolute terms in Figure 4c, and relative to the GT in Figure 4e). The face-level co-occurrences are poor – with an average error from GT of 48%. For instance, Penny and Leonard, whose romance is a main story-line, are shown to co-occur in only 3% of the videos vs the GT 9%. Furthermore, the GT shows that this is the second most commonly occurring pair; nevertheless, the face-level annotations fail to pick up that it is significant relative to other pairs. This is expected as often one or more characters do not show their face when appearing together (Figure 1). Hence, any co-occurrences predicted from the face-level are a limited foundation for story understanding.

For the second question, we cluster with *MuHPC* and assign each cluster to the character that appears most within it (Figures 4b, 4d). We observe that these predictions are very close to the GT, with an average error of just 3% (Figure 4d). This impressive result shows that the presence of each character, their co-occurrence and hence their possible interactions can be found completely automatically and accurately using our proposed method. This provides a rich and informative foundation for story understanding.

The assignment of character names to clusters can be automated by combining *MuHPC* with methods that focus on the automated labelling of face-tracks with names [6, 16].

6. Conclusions

In this work we propose *MuHPC*, a novel method for multi-modal person-clustering in videos. For evaluation we introduced *VPCD*, the largest and most diverse dataset of its kind. We showed that using all available video cues is essential for person-clustering, leading to significant improvements on *VPCD*, and to state-of-the-art performance for face-clustering. Importantly, we demonstrated that *MuHPC* allows each character appearance and co-occurrence to be predicted completely automatically and accurately. We hope this can support downstream story understanding tasks such as the learning of relationships [34]. *MuHPC* has intriguing benefits for creative video editing/understanding, such as the automated collation of character-based "highlight reels" *i.e.* scenes containing a certain two character's interactions, or one character's story-line. *MuHPC* can also serve as an essential foundation for methods of video story understanding.

References

[1] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, Jesse Zhang, Eliot Godard, Lukas L. Diduch, A. F. Smeaton, Yyette Graham, Wessel Kraaij, and Georges Quénot. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search retrieval. *ArXiv*, abs/2009.09984, 2019. 3

[2] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, 2020. 3

[3] Martin Bauml, Makarand Tapaswi, and Rainer Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *Proc. CVPR*, 2013. 2, 3

[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 *Proc. CVPR*, 2004. 2

[5] P. Bojanowski, F. Bach, , I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *Proc. ICCV*, 2013. 2

[6] Andrew Brown, Ernesto Coto, and Andrew Zisserman. Automated video labelling: Identifying faces by corroborative evidence. In *MIPR*, 2021. 2, 8

[7] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proc. CVPR*, 2018. 6

[8] 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. 5, 6

[9] 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. In *INTERSPEECH*, 2020. 6

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 6

[11] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Unsupervised metric learning for face identification in tv video. In *Proc. ICCV*, 2011. 1, 2, 3

[12] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *Proc. CVPR*, 2009. 3

[13] T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking pictures: Temporal grouping and dialog-supervised person recognition. In *Proc. CVPR*, 2010. 2

[14] Renato Cordeiro de Amorim. Constrained clustering with minkowski weighted k-means. In *CINTI*, 2012. 2

[15] Philippe Ercolessi, Hervé Bredin, and Christine Sénac. Stoviz: story visualization of tv series. In *Proc. ACMMM*, 2012. 3

[16] Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC*, 2006. 2, 3, 5, 6, 8

[17] Mark Everingham, Josef Sivic, and Andrew Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 2009. 2

[18] Andrew W. Fitzgibbon and Andrew Zisserman. On affine invariant clustering and automatic cast listing in movies. In *Proc. ECCV*, 2002. 1, 2

[19] Esam Ghaleb, Makarand Tapaswi, Ziad Al-Halah, Hazim Kemal Ekenel, and Rainer Stiefelhagen. Accio: A data set for face track retrieval in movies across age. In *Proc. ICMR*, 2015. 3

[20] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 6

[21] Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming tv characters by watching and analyzing dialogs. In *Proc. WACV*, 2016. 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 6

[23] Yue He, Kaidi Cao, Cheng Li, and Chen Change Loy. Merge or not? learning to group faces via imitation learning. In *AAAI*, 2018. 2

[24] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proc. CVPR*, 2003. 2

[25] Minh Hoai and Andrew Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *Proc. ACCV*, 2014. 7

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. CVPR*, 2018. 6

[27] Qingqiu Huang, Wentao Liu, and Dahua Lin. Person search in videos with one portrait through visual and temporal links. In *Proc. ECCV*, 2018. 3, 6

[28] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *Proc. ECCV*, 2020. 3, 6

[29] Raymond Austin Jarvis and Edward A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Trans. Computers*, 1973. 2, 3

[30] 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 *Proc. ICCV*, 2017. 1

[31] Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. Person recognition in personal photo collections. In *Proc. ICCV*, 2015. 3

[32] Vicky Kalogeiton and Andrew Zisserman. Constrained video face clustering using 1nn relations. In *Proc. BMVC*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[33] M. Köstinger, P. Wohlhart, P. Roth, and H. Bischof. Learning to recognize faces from videos and weakly relatedinformation cues. In *AVSS*, 2011. 2

[34] Anna Kukleva, Makarand Tapaswi, and Ivan Laptev. Learning interactions and relationships between movie characters. In *Proc. CVPR*, 2020. 3, 8

[35] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proc. CVPR*, 2014. 3, 6

[36] Wei-An Lin, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. Deep density clustering of unconstrained faces. In *Proc. CVPR*, 2018. 2

[37] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004. 4

[38] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge

university press, 2008. 6

[39] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman. LAEO-Net: revisiting people Looking At Each Other in videos. In *Proc. CVPR*, 2019. 3

[40] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman. LAEO-Net++: revisiting people Looking At Each Other in videos. In *IEEE PAMI*, 2020. 8

[41] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Learnable pins: Cross-modal embeddings for person identity. In *Proc. ECCV*, 2018. 4

[42] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Seeing voices and hearing faces: Cross-modal biometric matching. In *Proc. CVPR*, 2018. 4

[43] Arsha Nagrani and Andrew Zisserman. From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. In *Proc. BMVC*, 2017. 2, 5, 6

[44] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE PAMI*, 2017. 2

[45] Alexey Ozerov, Jean-Ronan Vigouroux, Louis Chevallier, and Patrick Pérez. On evaluating face tracks in movies. In *Intl. Conf. Image Proc.*, 2013. 3

[46] Omkar M. Parkhi, Esa Rahtu, and Andrew Zisserman. It's in the bag: Stronger supervision for automated face labelling. In *ICCV Workshop: Describing and Understanding Video & The Large Scale Movie Description Challenge*, 2015. 2

[47] Johann Poignant, Hervé Bredin, and Claude Barras. Multimodal person discovery in broadcast tv: lessons learned from mediaeval 2015. *Multimedia Tools and Applications*, 2017. 2

[48] Deva Ramanan, Simon Baker, and Sham Kakade. Leveraging archival video for building face datasets. In *Proc. ICCV*, 2007. 2

[49] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *Proc. ECCV*, 2014. 2

[50] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proc. CVPR*, 2020. 3

[51] Chandan Reddy and Bhanukiran Vinzamuri. *A Survey of Partitional and Hierarchical Clustering Algorithms*. 2018. 3

[52] Veith Röthlingshöfer, Vivek Sharma, and Rainer Stiefelhagen. Self-supervised face-grouping on graphs. In *Proc. ACMMM*, 2019. 2, 7

[53] Anindya Roy, Camille Guinaudeau, Hervé Bredin, and Claude Barras. Tvd: a reproducible and multiply aligned tv series dataset. In *LREC*, 2014. 2, 3, 5, 6

[54] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proc. CVPR*, 2019. 2, 3, 4, 7, 8

[55] 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. 2, 5

[56] 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. 2

[57] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Clustering based contrastive learning for improving face representations. *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2020. 2, 5, 7

[58] Josef Sivic, Mark Everingham, and Andrew Zisserman. "Who are you?" – learning person specific classifiers from video. In *Proc. CVPR*, 2009. 2

[59] Josef Sivic, C. Larry Zitnick, and Rick Szeliski. Finding people in repeated shots of the same scene. In *Proc. BMVC*, 2006. 3

[60] Krishna Somandepalli, Rajat Hebbar, and Shrikanth Narayanan. Multi-face: Self-supervised multiview adaptation for robust face clustering in videos. *arXiv preprint arXiv:2008.11289*, 2020. 5

[61] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "knock! knock! who is it?" probabilistic person identification in tv-series. In *Proc. CVPR*, 2012. 2

[62] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Storygraphs: visualizing character interactions as a timeline. In *Proc. CVPR*, 2014. 3

[63] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *Proc. ICCV*, 2019. 1, 2, 4, 5, 6, 7, 8

[64] 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 *Proc. ICVGIP*, 2014. 1, 2

[65] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *Proc. CVPR*, 2018. 3

[66] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proc. ICML*, 2001. 2

[67] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proc. CVPR*, 2018. 3

[68] P. Wohlhart, M. Köstinger, P. M. Roth, and H. Bischof. Multiple instance boosting for face recognition in videos. In *DAGM-Symposium*, 2011. 2

[69] Baoyuan Wu, Siwei Lyu, Bao-Gang Hu, and Qiang Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *Proc. ICCV*, 2013. 2

[70] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. Constrained clustering and its application to face clustering in videos. In *Proc. CVPR*, 2013. 1, 2

[71] Jiangyue Xia, Anyi Rao, Qingqiu Huang, Linning Xu, Jiangtao Wen, and Dahua Lin. Online multi-modal person search in videos. In *Proc. ECCV*, 2020. 3

[72] Shijie Xiao, Mingkui Tan, and Dong Xu. Weighted block-sparse low rank representation for face clustering in videos. In *Proc. ECCV*, 2014. 2

[73] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Utterance-level aggregation for speaker recognition in the wild. In *Proc. ICASSP*, 2019. 6

[74] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *Proc. CVPR*, 2015. 3

[75] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Joint face representation adaptation and clustering in videos. In *Proc. ECCV*, 2016. 2

[76] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong

Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proc. ICCV*, 2015. 3, 6

[77] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proc. ICCV*, 2017. 3, 6

Face, Body, Voice: Video Person-Clustering with Multiple Modalities Supplementary Material

Andrew Brown[1], Vicky Kalogeiton[1,2], and Andrew Zisserman[1]
[1]VGG, Dept. of Engineering Science, University of Oxford. [2]LIX, École Polytechnique, CNRS, IP Paris
{abrown, az}@robots.ox.ac.uk, vicky.kalogeiton@lix.polytechnique.fr
https://www.robots.ox.ac.uk/~vgg/data/Video_Person_Clustering/

Contents

1. Broader Impact 1

2. Supplementary Video Contents 1

3. *VPCD* Details 2
3.1. Annotation Process 2
3.2. Feature Extraction 2
3.3. *VPCD* Voice-Track Statistics 2

4. Implementation Details 2

5. Metrics 3

6. Qualitative Results 3

7. Modality Analysis 5

8. Person-Clustering Results 5
8.1. Per-Stage Analysis 6
8.2. Oracle Clusters Results 6
8.3. Clustering on Multiple Program Sets Simultaneously 7

9. Face-Clustering Results 8

10 Parameter Selection & Sweeps 8
10.1 Nearest Neighbor Distance Threshold 9
10.2 Automatically Learnt Hyper-Parameters . . . 10

1. Broader Impact

Video Person-Clustering is an appealing topic in Computer Vision, with many downstream applications such as story understanding, video navigation, and video organisation. A successful person-clustering framework (such as that presented in this work) takes a significant step towards realising these applications by alleviating the tremendous annotation cost that would otherwise be necessary.

For all potential impacts and applications of video person-clustering, it is essential that the datasets that methods are evaluated on are representative of the real-world in which they (or their downstream applications) may be deployed [13]. This is essential if the research is to be accessible by different communities around the world. A representative dataset can accurately foreshadow and ultimately prevent any algorithmic discrimination on specific demographic groups. Previous person-clustering datasets (which focused on the narrower task of face-clustering) were non-representative of most demographic groups. To this end, in this work we presented *VPCD*, which represents a wide and diverse range of characters, and so is more representative of the diversity in the real-world.

The person-clustering task aims at recognising and clustering identities. Re-identifying people in the real-world generally poses a threat to their privacy, and could carry risks if used inappropriately. In *VPCD* however, the identities are all actors playing the part of characters. This is not private data, and none of the videos have been obtained from social media or search engines. All videos in *VPCD* are in fact from public films and television material.

2. Supplementary Video Contents

Three videos are included with this supplementary material. Here, we explain what is contained in each of them.

The first is titled, "Story_Understanding". This video highlights the advantages of the new task of multi-modal video person-clustering, over the established, more limited task of face-clustering. The video visualises the amount of information that is used by these two tasks. For the face-clustering task, only information from visible faces is used. This omits important information such as characters viewed from behind, or from the audio track. This limits the utility of the resulting clusters for downstream applications such as story understanding. The multi-modal person-clustering task on the other hand uses all available cues (*i.e.* face, body and voice). Clearly, a person-level understanding is essential

for downstream applications of grouping-by-identity such as story understanding.

The second is titled "VPCD_Contents". This video visualises the different annotations provided in *VPCD*, namely the face-tracks, body-tracks (from front and behind), and voice-tracks. The video shows clips containing example annotations from all 6 program sets in *VPCD*. Every face-body pair belonging to the same person-track is drawn with the same color. Note, *VPCD* covers a diverse set of characters in a variety of scenes (*e.g*. including dark scenes in Sherlock, Hidden Figures), various viewpoints and over-the-shoulder shots.

The third video, titled "MuHPC_Results" shows a selection of *MuHPC* person-clustering results from *VPCD*. In each clip, tracks are marked with a unique cluster ID number and colour, which signify which cluster they belong to. Particularly of note are the multiple backs of people that are clustered correctly *i.e*. Ross (cluster 2) in the first clip from Friends, Penny (cluster 3) in the second clip from TBBT, and the multiple backs in the dark scene from Buffy. Impressively, the very small Chandler, Joey and Ross tracks (clusters 1,0,2, respectively) at the back of the shot at the end of the Friends clip, are correctly clustered with other tracks of the same character.

3. *VPCD* Details

Here, we give additional details on the annotation (Section 3.1) and feature extraction (Section 3.2) process for the body-tracks in *VPCD*. These sections are complementary to Sections 4.2 & 4.3 in the main manuscript. We then give further statistics and details of the voice-tracks in *VPCD* (Section 3.3).

3.1. Annotation Process

Here, we provide additional details for the body-track annotation in *VPCD*. To set the scene, we have body-tracks computed for all program sets in *VPCD*. The task at this stage is to annotate the body-tracks with the names of the characters that are annotated in the face-tracks.

The body-tracks fall into two categories, which are annotated separately. (1) The body-track shows the person from the front and contains a visible, annotated face. For these cases we automatically label the body-tracks by making assignments to labelled face-tracks. Within each shot, the assignment is done using the Hungarian Algorithm [8] with a cost function of the spatial intersection over union (IOU) between face and body-tracks in the frames that they co-occur. If there are more body-tracks than face-tracks, then a body-track can not be assigned, and vice-versa. In 95% of cases this association is trivial and the assignment proceeds automatically. Where multiple assignment costs for the same face-track are below a threshold, indicating that the assignment was non-trivial, we instead make the assignments manually. (2) The body-track does not contain a visible face, *i.e*. the back is turned to the camera. We manually annotate all of these cases throughout each video. On average, 10-15% of body-tracks correspond to manually labelled bodies from behind.

3.2. Feature Extraction

Here, we describe in more detail the feature extraction process for the body-tracks.

Features are extracted from each of the body-tracks using a ResNet50 architecture [5]. Our goal is to train the body features to discriminate identity based on the highly discriminative clothing that people are wearing. We train a ResNet50 on the CSM dataset [6], which contains identity-labelled body detections from movies. This dataset contains the same label for all body detections of each identity, regardless of their clothing. Instead, we decompose the samples for each class (identity) in CSM into sub-classes containing images of the same identity in the same outfit. Our assumption is that if two detections occur close-by temporally within the same movie, then the person is likely to be wearing the same clothing. Each body detection is annotated with the shot that the detection is found in. We cluster the body detections in each class according to their temporal location, resulting in several sub-classes for each identity, where they are wearing the same clothing. We train the model in a contrastive manner using the Smooth-AP loss from [2]. For the network to be variant to both identity and clothing, we sample positives from the same identity wearing the same outfit, and negatives from different identities.

3.3. *VPCD* Voice-Track Statistics

Here, we give further details and statistics for the voice-tracks in *VPCD*. In total, there are 27,163 voice-tracks in *VPCD* (Table 1). This includes annotations for the 'laughter' track from the live studio audience in TBBT and Friends, and additionally laughter from each character in all program sets. Features, and the associated annotations for all of these voice-tracks are provided for future research use with *VPCD*. The distribution of lengths of these voice-tracks is shown in Figure 1. These figures for the number of voice-tracks are different to those provided in Table 1 in the main manuscript. *MuHPC* implements a pre-processing step on the voice-tracks, such that only the most identity-discriminating voice-tracks are used in the clustering process (explained in Section 4).

4. Implementation Details

In this section, we give details on a pre-processing step for *MuHPC*, which aims to remove voice-tracks that might not be identity-discriminating from the clustering process. Some of the voice-tracks in *MuHPC* are not used, due to overlap between multiple voice-tracks, or due to them being

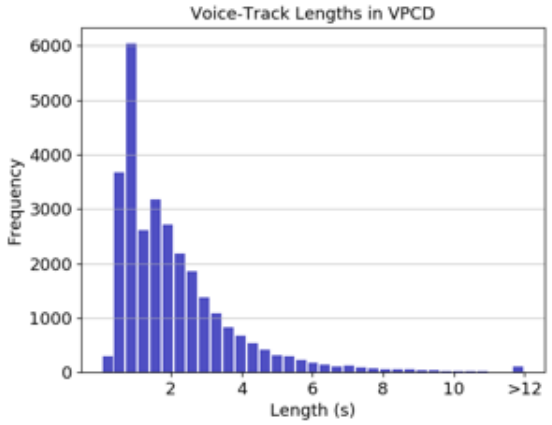


Figure 1: **Voice-track lengths in *VPCD*.** The distribution of all voice-track lengths in *VPCD*.

	TBBT	Buffy	Sherlock	Friends	HF	ALN	Total
All Annotations	2,035	4,339	4,025	11,321	2,060	2,036	**27,163**
Filtered	1,047	1,835	1,615	3,961	404	303	**9,165**

Table 1: **Voice-Track statistics in *VPCD*.** The number of voice-tracks for each program set in *VPCD* both before and after a filtering step (Section 3.1). All Annotations – the total voice-track annotations provided with *VPCD*. Filtered – the total voice-track annotations used by our person-clustering method, after ignoring short and overlapping tracks (same as Table 1 in main manuscript). Total – the summation over all six program sets.

too short. Here, we explain this process, and provide statistics on how many voice-tracks are ignored at this stage (Table 1). First, the temporal overlap between multiple voice-tracks. Our goal here is to use the voice-track features as a discriminative signal for identity. If multiple voice-tracks from different identities have large temporal overlap, then the resulting features will be very similar, and they will not provide a good identity-discriminating signal. We choose to ignore any voice-tracks that have 20% overlap with a different voice-track. Second, the temporal length of the voice-tracks. As shown in [16], there is a strong positive correlation between the discriminative capabilities of voice-track features and the length of the voice-track. In order to maximise the discriminativeness of the voice-track features, we ignore those that are less than 1 second in length. Table 1 shows the total number of voice-track annotations in *VPCD* before ("All Annotations") and after these steps ("Filtered").

5. Metrics

As mentioned in Section 5 in the main manuscript, for each dataset in *VPCD*, we use Weighted Cluster Purity (WCP) and Normalized Mutual Information (NMI). Furthermore, we introduce the metrics of Character Precision and Recall. Here, we describe in more detail the WCP and NMI metrics and give some motivation behind the proposed Character Precision and Recall (CP, CR).

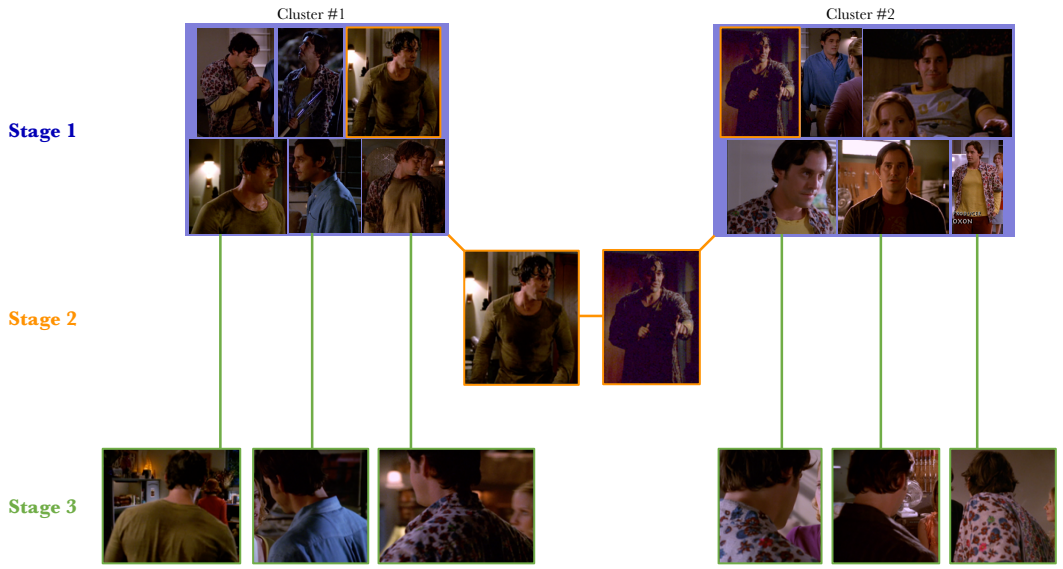
Weighted Clustering Purity (WCP). WCP weights the purity of a cluster by the number of samples belonging in it; to compute purity, each cluster c containing n_c elements is assigned to the class which is most frequent in the cluster. WCP is highest at 1 when within each cluster, all samples are from the same class. For a given clustering, C, with N total tracks in the video: $WCP = \frac{1}{N}\sum_{c\in C} n_c \cdot purity_c$.

Normalized Mutual Information (NMI) [9]. NMI measures the trade-off between clustering quality and number of resulting clusters. Given class labels Y and cluster labels C, $\mathrm{NMI}(Y,C) = 2\frac{I(Y;C)}{H(Y)+H(C)}$, where $H(.)$ is the entropy and $I(Y;C) = H(Y) - H(Y\backslash C)$ the mutual information.

Character Precision and Recall (CP, CR). We introduce Character Precision (CP) and Recall (CR), two metrics computed using the ground truth number of clusters. CP is the proportion of tracks in a cluster that belong to its assigned character, while CR is the proportion of that character's total tracks that appear in the cluster. The assignment is done using the Hungarian algorithm [8] by using CR as the cost function. Note that this assignment is unique, *i.e.* two characters cannot be assigned to the same cluster. We measure CP and CR and report results averaged across all characters. Our motivation is that the standard metrics are weighted according the number of samples in each cluster, thus disproportionately favouring frequently appearing characters and disregarding tail distributions. Instead, similar to character AP [10], CP and CR weight all characters equally. Similar to the Hungarian matching accuracy used in [1, 15], CP and CR are computed using the ground truth number of clusters. Thus, they measure complementary information to WCP and NMI, which do not have access to this information.

6. Qualitative Results

Further qualitative examples of the clustering process for characters in two of the program sets in *VPCD* are shown in Figure 2. In both cases, Stage 1 is shown to produce high-precision clusters of the character. The face alone cannot confidently merge these clusters, due to each cluster containing different views of the same character (*e.g.* frontal and profile). These clusters are merged via speaking person-tracks, using the multi-modal bridges of Stage 2. Back views of the same character are then merged into the clusters in Stage 3. The resulting clusters contain differing views of the same character, with varying pose, lighting conditions, and camera viewpoints, all while maintaining high precision.



(a) **Clustering Process of *MuHPC* for a character in Buffy.** Stage 1 produces high-precision clusters. Cluster #1 contains mainly profile and downwards-facing views of the character, while Cluster #2 contains frontal facing views. Both clusters contain very different clothing and body poses. The face modality alone can no longer confidently merge these clusters. Stage 2 merges the two clusters using multi-modal bridges between a speaking person-track from each cluster. Stage 3 then merges back views into these clusters via body features. Back views of the character are merged via frontal appearances in nearby shots where the character is wearing the same clothing.



(b) **Clustering Process of *MuHPC* for a character in Sherlock.** Stage 1 produces high-precision clusters. Cluster #1 contains mainly frontal face views, while Cluster #2 contains profile face views. Both clusters contain very different lighting conditions, body poses; and camera-views of the same character. Stage 2 merges the two clusters where the face alone could not, by using multi-modal bridges between a speaking person-track from each cluster. Stage 3 then merges back views into these clusters via body features. Back views of the character (both full-body, and over-the-shoulder views) are merged via frontal appearances in nearby shots where the character is wearing the same clothing.

Figure 2: **Clustering Process of *MuHPC*.** For two program sets from *VPCD*, (a)-Buffy, and (b)-Sherlock, we show the clustering process for one of the principal characters.

	Modality			Protocol	Average			
	F	B	V		WCP	NMI	CP	CR
$MuHPC_{body}$		✓		AT	60.6	46.9	63.4	48.1
$MuHPC_{voice}$			✓	AT	71.0	67.9	54.6	50.3
$MuHPC_{face}$	✓			AT	**93.4**	**89.4**	**93.0**	**90.2**
$MuHPC_{body}$		✓		OC	58.1	43.7	50.6	44.8
$MuHPC_{voice}$			✓	OC	77.5	70.4	58.1	55.2
$MuHPC_{face}$	✓			OC	**91.7**	**87.2**	**84.7**	**81.9**

Table 2: **Person-Clustering Results on *VPCD* after Stage 1 – Clustering only speaking person-tracks.** We report the averaged metrics for both AT and OC protocol, averaged across all program sets. Every experiment shown is clustering only a subset of the person-tracks that contain all three modalities (face, body and voice) in order to isolate the clustering performance when each modality is used alone. The three reported methods, $MuHPC_{body}$, $MuHPC_{voice}$, $MuHPC_{face}$, use a different modality as the single modality in Stage 1 (body, voice and face, respectively). The numbers reported are taken after Stage 1.

7. Modality Analysis

In this section, we provide further analysis into the discriminative capabilities of each of the three modalities used in *MuHPC* (face, body and voice). In Stage 1 of *MuHPC*, high-precision clusters are created using just the face modality, as it is the most discriminative of the three. Here, we justify this by instead using the other modalities in Stage 1. Table 2 shows results averaged across all program sets in *VPCD* for both AT and OC protocol, when each of the available modalities are used in Stage 1 (termed $MuHPC_{body}$, $MuHPC_{voice}$; and $MuHPC_{face}$). Next, we explain some experimental details, and then analyse these results.

For fair comparison between $MuHPC_{body}$, $MuHPC_{voice}$; and $MuHPC_{face}$, we cluster the same person-tracks in each of the experiments. This limits the experiments to person-tracks with all three available modalities *i.e*. talking person-tracks with a visible face. To isolate the role of each of the modalities, we report clustering performance after Stage 1. Similarly to τ_f^{tight} in *MuHPC*, for these experiments we learn nearest neighbour distance thresholds for each modality on the *VPCD* val. set.

As shown in Table 2, only the face modality can be reliably used in Stage 1 to produce high-precision clusters, as reflected by the high values for WCP in both protocol. This justifies the use of the face modality in Stage 1 of *MuHPC*. This is understandable, as different identities can sound the same when expressing similar emotions (*e.g*. anger, sadness), and bodies from different identities can look very similar when wearing similar clothing. According to WCP and NMI, $MuHPC_{voice}$ produces better clustering performance than $MuHPC_{body}$, indicating that the voice modality is better at discriminating identity than the body modality.

8. Person-Clustering Results

In this section, we provide extensive analysis of the person-clustering results obtained by *MuHPC* as well as results for an additional experiment. First, we explore the impact of Stages 1 and 2 of *MuHPC* on some episodes from the Friends program set in *VPCD* (Section 8.1). Second, we provide further person-clustering results from *MuHPC* on *VPCD* using the OC protocol. Third, we examine the results when clustering tracks from all program sets in *VPCD*, concatenated by their research order of broadcast (Section 8.3).

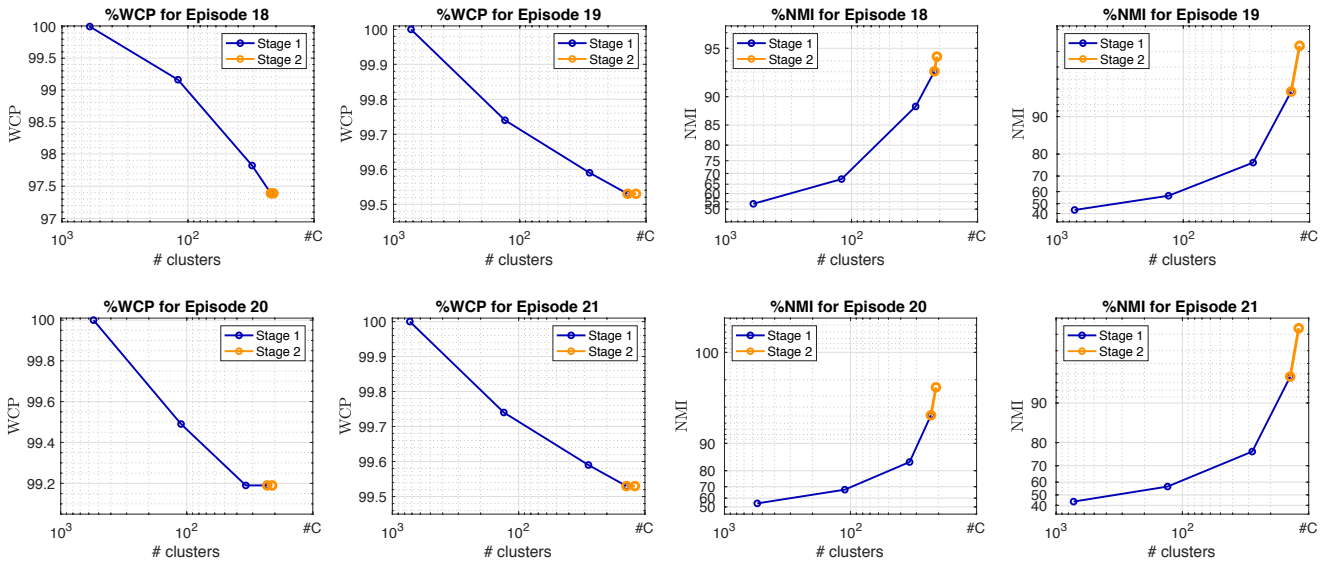


Figure 3: **Stage 1 and Stage 2 Person-Clustering results from the program set, Friends.** %WCP and %NMI for episodes of Friends from *VPCD*, for the Automatic Termination protocol (AT). The blue line illustrates the results after Stage 1, while the orange one illustrates the results after Stage 2, *i.e.* bridging clusters by exploiting the voice modality. *#C* is the ground truth number of clusters for each episode.

8.1. Per-Stage Analysis

We examine the effects of Stages 1 and 2 (Section 3 in the main manuscript) on the performance of *MuHPC* on episodes from the Friends program set in *VPCD*. To this end, we plot in Figure 3 the %WCP and %NMI results over the number of clusters after each partition of the method for four episodes. Each circle in the plot displays the partition (*i.e.* showing the number of clusters of the resulting partition and the corresponding metric value). The blue lines and circles represent the clustering process at Stage 1 of *MuHPC*, while the orange ones display the Stage 2 results.

We observe that in most cases after the first partition (first blue dot) the WCP maintains high values (above 99%). While Stage 1 progresses, the WCP drops only by a small margin (*i.e.* less than 1% in most cases), whereas the NMI increases significantly (*i.e.* up to +50%). This validates that Stage 1 indeed results in high-precision clusters, as the purity (indicated by WCP) is not compromised, and also the NMI increases.

The orange dots signify the additional partition from Stage 2. Stage 2 consistently and significantly increases the NMI of the resulting clusters (*i.e.* by up to 5%), without sacrificing their purity (WCP remains constant). This indicates that Stage 2 bridges high-precision clusters of the same identity, thus retaining the high WCP, while decreasing the identity overlap between clusters.

8.2. Oracle Clusters Results

Table 3 gives person-clustering results for the OC protocol. The experiments, ablation studies and baselines are the same as those used for the AT protocol, and explained in Section 5.1 of the main manuscript. Similarly to the AT protocol, *MuHPC*– significantly outperforms both baselines across all metrics and program sets. *MuHPC* gives a further boost when averaged across all program sets. The voice modality provides comparably less of a performance boost in the OC protocol (here) relative to the AT protocol (Table 2 in the main manuscript). This is due to the Oracle Cluster protocol (OC), which forces the clusters to merge beyond the automatic termination point until the ground truth number of clusters is reached. Next, we explain this in further detail.

MuHPC automatically stops clustering when the features within each cluster can no longer confidently be used to discriminate between clusters of the same identity. To reach the oracle number of clusters, the clusters are merged in a non-discriminative way. In this case, this reverses the positive impact of the voice modality (seen in Table 2 in the main manuscript) by merging the new clusters incorrectly until the oracle number of clusters is reached. This opens possibilities for future research into more effective ways of reducing to the ground truth number of clusters. The Automatic Termination protocol is the more realistic setting for real-world deployment of person-clustering algorithms on videos with unknown numbers of characters.

#	Modality			TBBT				$\#C_s$= 130	Buffy				$\#C_s$= 165	Sherlock				$\#C_s$= 50	Friends				$\#C_s$= 239	Hidden Figures				$\#C_s$= 10	About Last Night				$\#C_s$= 24	Average				$\#C_s$= 618
	F	B	V	WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR		WCP	NMI	CP	CR	
B-ReID		✓		66.2	54.9	11.4	33.0		52.0	48.8	40.6	24.6		60.3	24.0	10.7	36.0		62.8	56.0	49.8	56.4		33.7	10.5	17.6	31.7		27.3	17.9	19.9	19.3		50.4	35.4	25.0	33.5	
B-C1C	✓	✓		91.7	79.1	54.4	55.9		74.5	62.7	46.8	44.7		77.1	44.4	33.2	43.3		88.0	82.4	74.5	78.9		69.5	51.8	29.7	46.4		73.1	64.8	55.7	53.8		79.0	64.2	49.1	53.8	
MuHPC-	✓			94.3	85.8	84.1	81.8		81.1	68.0	76.2	76.3		86.79	56.87	74.8	**69.3**		90.0	76.6	90.8	82.8		**85.7**	**77.3**	**76.7**	**56.7**		**97.9**	91.4	98.9	86.9		89.3	76.0	83.6	75.6	
MuHPC$_v$	✓		✓	94.3	85.8	84.1	81.8		81.1	68.5	76.2	75.8		86.1	62.3	72.8	68.8		89.8	77.3	89.6	84.6		85.7	77.3	76.7	56.7		97.8	**91.9**	98.9	87.0		89.3	76.4	83.4	75.5	
MuHPC$_b$	✓	✓		**97.7**	**93.9**	**86.9**	**83.8**		86.9	76.9	**80.0**	**79.1**		**87.1**	57.5	**74.9**	66.8		**94.2**	84.6	**95.7**	86.0		85.6	77.1	76.6	56.7		97.8	91.2	98.9	86.9		**91.5**	80.2	**86.0**	77.0	
MuHPC	✓	✓	✓	97.7	93.9	86.9	83.8		**86.9**	**77.6**	79.8	78.5		86.4	**63.0**	73.1	68.7		94.0	**85.5**	94.6	**88.1**		85.6	77.1	76.6	56.7		97.8	91.6	**98.9**	**87.0**		91.4	**81.5**	85.0	**77.1**	

Table 3: **Person-Clustering Results on *VPCD*.** For each program set, each metric is averaged across all episodes. OC protocol. The 'Average' column reports averaged metrics across all six program sets. $\#C_s$ is the sum of ground truth clusters across each episode. We report two strong baselines (B-ReID, B-C1C, Section 5.1 in main manuscript) and an ablation on the modalities used. Keys: F-face, B-body, V-voice. *Modality*: used modalities.

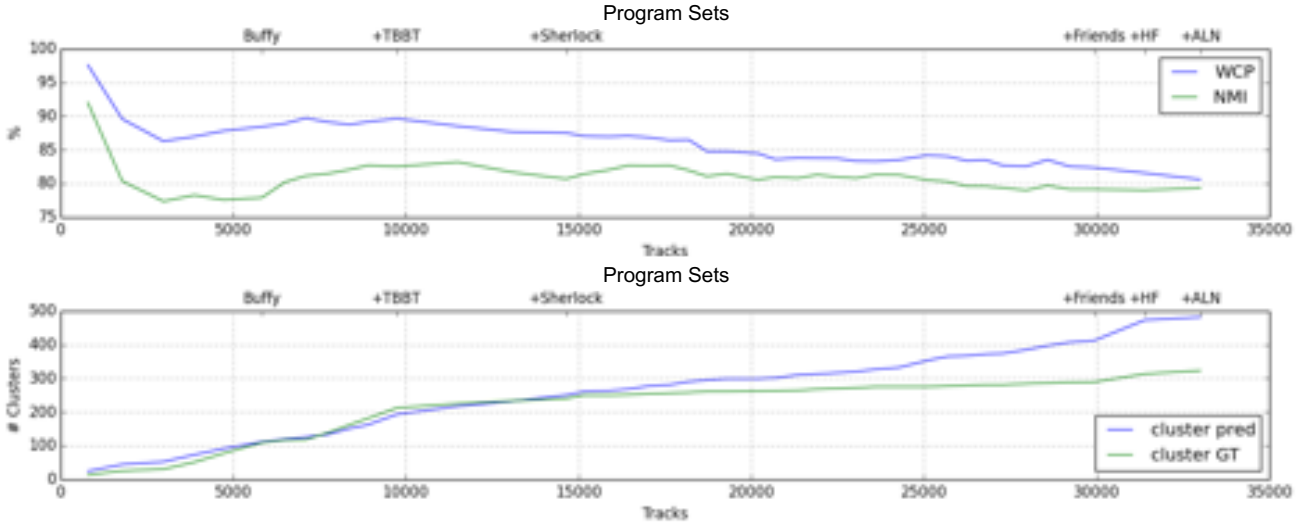


Figure 4: **Person-Clustering Results when clustering multiple program sets simultaneously.** Incrementally, more and more tracks are considered by adding different program sets together. There are discrete data points for each time the tracks from an additional episode or movie are added. Each data point considers the total cumulative number of tracks up to that point. All experiments are for the Automatic Termination (AT) protocol for person-clustering for *MuHPC*. Top: The WCP and NMI measurements. Bottom: The total predicted number of clusters (cluster pred), measured against the ground truth number of clusters (cluster GT). Note that "cluster GT" is different to $\#C_s$ in the main manuscript. $\#C_s$ is the summed number of ground truth clusters (number of characters) across multiple episodes. For example, episodes 1 and 2 of Sherlock have 13 and 22 ground truth clusters, respectively. In this case, $\#C_s = 35$. However, some characters appear in both episodes, such as "John" or "Sherlock". Instead, "cluster GT" is the total number of *unique* ground truth characters and therefore clusters across multiple episodes. For the same example of episodes 1 and 2 of Sherlock, "cluster GT" is equal to 31, as 4 characters feature in both episodes.

8.3. Clustering on Multiple Program Sets Simultaneously

In this section, we present results for the person-clustering task when clustering tracks from multiple program sets simultaneously. In the main manuscript, all experiments are conducted on individual program sets from *VPCD*. Here, we cluster tracks from multiple program sets at the same time. In detail we incrementally consider additional episodes and movies from each of the program sets. Results for the WCP, NMI and the number of predicted clusters against the ground truth number of characters for the AT protocol for person-clustering are shown in Figure 4. The order with which program sets are added to the clustering experiment is in line with the timing of their first use in Computer Vision research (*i.e.* first Buffy [3], followed by TBBT [11], then Sherlock [10] and so on). Episodes within each of the TV-shows are added chronologically (starting with the first episode in the program set).

Impressively, Figure 4 shows that when clustering all tracks from *VPCD* simultaneously, the WCP and NMI remain high at 80.6% and 79.3%, respectively. This indicates that most clusters have high purity, even with 323 different characters and over 30,000 tracks, over the visually disparate TV-shows and movies. As expected, these metrics drop as the total number of tracks increases, as the task becomes much more difficult. Until the introduction of tracks from episodes from Friends (14,642 tracks), the predicted number of clusters lies very close to the ground truth number of clusters. This indicates that *VPCD* is accurately predicting the number of different characters in the tracks. As the total

number of tracks increases, the predicted number of clusters diverges from the ground truth number, and *MuHPC* predicts more clusters than there are characters. This is in line with and partially explained by the combination of cannot-link constraints and decreasing WCP. As the purity of clusters decreases, the cannot-link constraints start preventing clusters containing tracks of the same identity from merging. This results in *MuHPC* automatically terminating the clustering when there are more clusters than characters. We observe similar results when adding datasets in different orders. Similar experiments for combining the TBBT and Buffy datasets for face-clustering are presented in [14].

9. Face-Clustering Results

Here, we give further analysis of the face-clustering results shown in Table 3 of the main manuscript (and repeated in Table 4). This is an extension of Section 5.3 in the main manuscript. In detail, the extra analysis concerns the automated termination (AT) criterion, and the relation of *MuHPC* to previous methods. To summarise Section 5.3 of the main manuscript, *MuHPC* significantly surpasses the performance of previous methods across all program sets, all metrics and both AT and OC protocol.

First, we analyse the AT protocol results. The goal of the AT protocol is to automatically terminate clustering and assess the quality of the resulting clusters. This is a realistic protocol for videos in-the-wild where the number of characters is unknown. Here, the number of predicted clusters, $\#C_p$, can be measured relative to the ground truth number of clusters, $\#C_s$. In all program sets, *MuHPC* predicts more clusters than the ground truth. This is because *MuHPC* prioritises high-precision. For TBBT, $\#C_p$ is very close to $\#C_s$ (168 vs. 130), and is in fact closer than the predictions of all previous methods. This is impressive seeing as the goal of BCL [14] is to predict the ground truth number of clusters. For the other program sets, $\#C_p$ is slightly further from $\#C_s$ than previous methods (*e.g.* a difference from $\#C_s$ of 36 for Sherlock vs. 25 for C1C [7]). We now give two reasons why despite this, the clusters from *MuHPC* are far more desirable than those from previous methods.

First, the clusters from *MuHPC* are far higher quality. It would be expected that when predicting more clusters than there are ground truth clusters, any method would achieve higher WCP. However, NMI is also significantly higher for *MuHPC* than previous methods (*e.g.* on average 9.8% higher than the best prior work across all program sets). Second, for downstream applications, it is far more useful to have many high-precision clusters, than few very low-precision clusters. The latter in this case requires a large amount of human labelling in order to correctly label the person-tracks from the clusters (a cluster property reflected by the *Operator Clicks Index* (OCI-k) [4] metric). Furthermore, a good way of measuring the utility of clusters for a downstream task

Method	protocol	TBBT				$\#C_s$ = 130	Buffy				$\#C_s$ = 165
		WCP	NMI	CP	CR	$\#C_p$	WCP	NMI	CP	CR	$\#C_p$
BCL [14]	AT	90.8	85.7	-	-	83	85.0	78.8	-	-	121
C1C [7]	AT	89.2	87.4	29.1	40.9	41	66.3	68.8	14.9	27.1	40
MuHPC–	AT	**99.4**	**97.8**	**87.8**	**88.6**	168	**96.1**	92.8	85.6	**85.5**	223
$MuHPC_v$	AT	**99.4**	**97.8**	**87.8**	**88.6**	168	**96.1**	**93.7**	**85.9**	84.8	221
Finch [12]	OC	90.8	80.5	46.1	44.2		82.9	75.3	49.6	41.0	
BCL [14]	OC	94.0	85.0	-	-		86.5	77.6	-	-	
C1C [7]	OC	95.3	84.5	54.9	57.3		88.1	79.1	58.1	55.4	
MuHPC–	OC	**99.1**	**97.4**	**79.3**	**83.0**		**95.6**	92.2	**72.3**	**73.8**	
$MuHPC_v$	OC	**99.1**	**97.4**	**79.3**	**83.0**		**95.6**	**93.1**	71.5	73.2	
		Friends				$\#C_s$ = 239	**Sherlock**				$\#C_s$ = 50
C1C [7]	AT	88.2	89.8	62.4	73.2	185	76.3	50.3	20.2	41.0	25
MuHPC–	AT	**98.7**	94.9	98.1	94.0	543	**86.7**	60.3	79.1	71.2	96
$MuHPC_v$	AT	98.4	**95.9**	97.7	**95.3**	522	86.3	**66.0**	**78.4**	**74.5**	86
Finch [12]	OC	92.2	89.9	85.2	85.6		81.6	58.6	**59.8**	56.8	
C1C [7]	OC	94.3	93.2	79.1	85.5		81.6	53.8	40.5	51.7	
MuHPC–	OC	96.3	92.7	89.0	88.8		84.0	56.5	55.4	59.9	
$MuHPC_v$	OC	**97.1**	**94.6**	**92.3**	**92.6**		**85.1**	**63.9**	59.6	**62.9**	

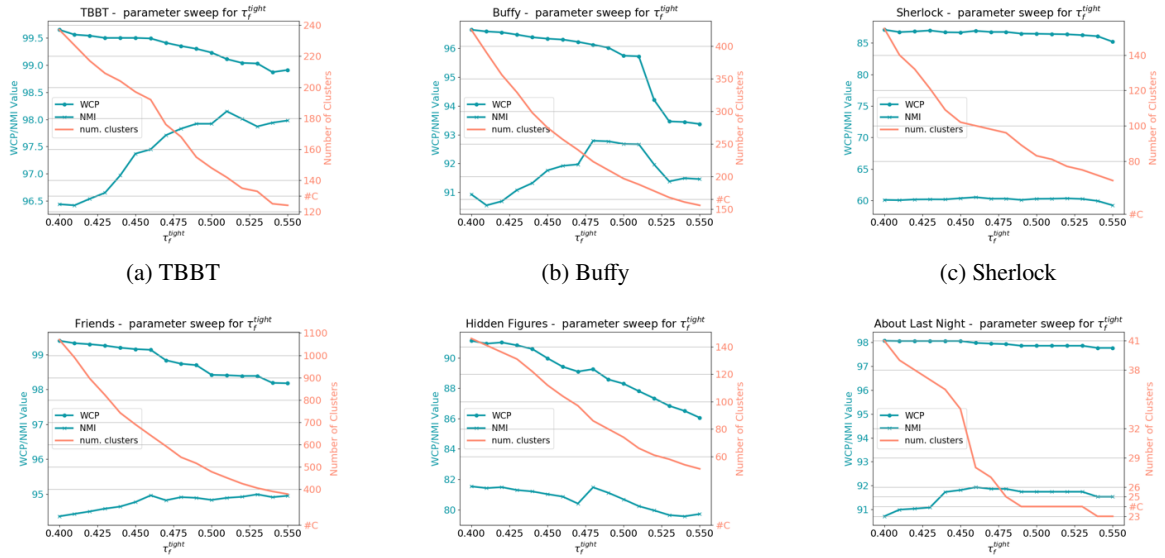
Table 4: **Face-Clustering Results.** Comparisons to previous state of the art on four program sets using only face-tracks with unknown (AT), and known (OC) number of clusters. We report metrics averaged over each episode in each program set, and the number of predicted clusters, summed over each episode ($\#C_p$). *MuHPC–* uses only face; $MuHPC_v$ uses the multi-modal bridges from voice and face. Where not reported in respective publications, numbers are computed using official implementations. Finch has no stopping criterion so results for AT are not reported.

is the character precision and recall metrics. These metrics assign each character uniquely to a cluster, and measure the resulting precision and recall of these pseudo-labels. *MuHPC* significantly achieves a CP and CR of 56.0% and 39.3% higher, respectively, than C1C across all program sets. This indicates that although prior work may predict a number of clusters closer to the ground truth than *MuHPC*, these clusters however are of almost no use for downstream applications, unlike the clusters from *MuHPC*.

Next, we discuss *MuHPC* in relation to previous methods. C1C continues using face to cluster even when there are large distances between clusters, and therefore degenerates in the later partitions, leading to lower WCP and NMI. Unlike BCL, *MuHPC* uses pre-trained features, thus alleviating the computational burden of training, allowing for greater generalisation, and as we demonstrate leading to better results. BCL uses the assumption that each identity occupies the same hyper-spherical volume in their learnt latent space. We argue that complex similarity structures and variation between faces of the same identity mean that they cannot be constrained to within fixed-radius hyper-spheres (BCL), even when training with this objective. Instead, *MuHPC* does not enforce such a constraint, and uses a nearest neighbour constraint with multi-modality to connect highly dissimilar tracks.

10. Parameter Selection & Sweeps

In this section, we give a parameter sweep for the nearest neighbour distance threshold τ_f^{tight} (Section 3.1 in main manuscript), and give further description and analysis on the automatic parameter selection method for τ_v^{loose} (Section 3.2 in main manuscript).



(a) TBBT (b) Buffy (c) Sherlock

(d) Friends (e) Hidden Figures (f) About Last Night

Figure 5: **Parameter sweep for τ_f^{tight} on the six program sets in *VPCD*.** For each program set, the NMI, WCP and number of clusters are plotted, for the Automatic Termination criterion, for varying values of τ_f^{tight}. We additionally show for each program set, the ground truth number of clusters, #C, marked on the Number of Clusters axis of each plot. For the numerical values of #C, we refer the reader to Table 2 in the main manuscript.

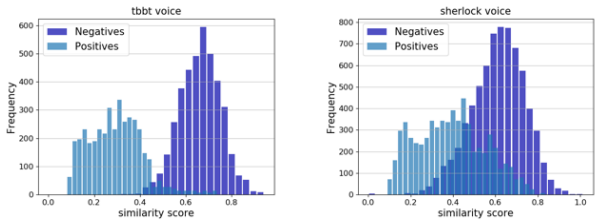
10.1. Nearest Neighbor Distance Threshold

Here, we give metrics across all program sets in *VPCD* for parameter sweeps on the nearest neighbour distance threshold, τ_f^{tight}. These are displayed in Figure 5. As detailed in the main manuscript, the value was chosen on the validation partition of *VPCD*. To isolate the role of τ_f^{tight}, all metrics are evaluated at the Automated Termination criterion, after Stage 1, and using only the face-track annotations. The metrics at the chosen value of $\tau_f^{\text{tight}} = 0.48$, are therefore equivalent to *MuHPC*– at AT protocol in Table 3 in the main manuscript. We notify the reader that in the main manuscript, it reads that $\tau_f^{\text{tight}} = 0.52$. This is incorrect, the value is $\tau_f^{\text{tight}} = 0.48$.

Across most program sets, the same relationship between the metrics and τ_f^{tight} is seen. Namely, as τ_f^{tight} increases, NMI increases, while WCP and the total number of clusters decreases. In more detail, as τ_f^{tight} increases, the maximum distance at which clusters can merge increases. This leads to more cluster merges before the automatic termination of Stage 1. This is reflected by the decreasing number of clusters at the termination point. Firstly, there is an increased likelihood of incorrect merges, where clusters depicting different identities merge together, leading to lower precision clusters, as shown by decreasing WCP. Increasing τ_f^{tight} also leads to more correct merges. This is reflected by the rising NMI, which shows that the identity overlap between clusters is decreasing. An increasing NMI can be interpreted as there being more correct merges than incorrect merges. In some program sets (*e.g.* Buffy, Sherlock), NMI starts to decrease as τ_f^{tight} increases, indicating that more incorrect merges are being made than correct merges.

In a window surrounding the learnt value of 0.48, the NMI and WCP are roughly stable at very high values across all program sets (high relative to the respective prior work on those program sets - see Table 3 in main manuscript). This demonstrates that this learnt parameter generalises well to the different program sets, that the face features are indeed universal; and that *MuHPC* is not particularly sensitive to this choice of parameter. The program sets in *VPCD* are highly visually disparate. These results therefore indicate that *MuHPC* could be simply and effectively used on any number of *different program sets* not in *VPCD*.

At the chosen value of $\tau_f^{\text{tight}} = 0.48$, often more clusters are predicted than the ground truth number (marked as #C in Figure 5). In some program sets, this is by just a small number (168 vs $\#C = 130$ for TBBT, 223 vs $\#C = 165$ for Buffy). There is a trade-off between obtaining a number of clusters similar to #C, and the precision of these clusters. Our design choice at Stage 1 is to produce clusters with very high-precision. Stage 2 leads to a further reduction of these clusters by using multiple modalities to merge clusters. A discussion in Section 9 explains why over-predicting the number of clusters is beneficial for downstream uses of the clusters.



(a) TBBT (b) Sherlock

Figure 6: **Voice similarities in two program sets from *VPCD*.** Here we show similarities between voices of the same identity (positives) and different identities (negatives). These are found via the cannot-link constraints (negatives) and the clusters from Stage 1 (positives and negatives). Similarities are computed via (1 minus cosine similarity). This process finds less positives than negatives, hence the frequency of the positives is scaled to match that of the negatives.

	TBBT	Buffy	Sherlock	Friends	HF	ALN
τ_v^{loose}	0.36	0.17	0.19	0.31	0.19	0.33

Table 5: **The automatically learnt values for τ_v^{loose} for the different program sets in *VPCD*.**

10.2. Automatically Learnt Hyper-Parameters

The values for the threshold on the voice similarities that are used in the multi-modal bridges, τ_v^{loose}, are learnt *automatically* for each of the audibly disparate program sets in *VPCD* (this is detailed in Section 3.4 in the main manuscript). Here, we give the values that are learnt for each program set, provide some analysis, and visualise the voice distances that the hyper-parameters were learnt from.

The values of τ_v^{loose} learnt automatically for the different program sets are given in Table 5. The voice distances between different identities are found via a combination of cannot-link constraints and the clusters from Stage 1. We observe that for some program sets these voice distances are quite high. This in turn leads to a relatively high value of τ_v^{loose} (*e.g.* TBBT, Friends). We additionally show the similarities between voices for the same identity (positives) and different identities (negatives) in Figure 6 for two program sets from *VPCD*.

A high value of τ_v^{loose} indicates that the characters all sounded different to the voice embedding network, and in turn the respective features from different speakers were able to be separated in the embedding space (Figure 6 - left). For the multi-modal bridges, this means that the voices from two speaking person-tracks can sound quite different and a bridge can still confidently be formed.

For other program sets, the voice distances between the different identities are quite low, and therefore τ_v^{loose} is also low (*e.g.* Buffy, Sherlock). In these cases, there are many similar sounding characters; hence, the voice embedding network cannot separate the embeddings from different identities well (Figure 6 - right). For the multi-modal bridges, this means that the voices from two speaking person-tracks must sound very similar for a bridge to still confidently be formed, as only then can the voice modality (together with the concurrent agreement from the face modality) be sure that it is the same person.

References

[1] Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *NeurIPS*, 2020. 3

[2] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *Proc. ECCV*, 2020. 2

[3] Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC*, 2006. 7

[4] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. 2009. 8

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2

[6] Qingqiu Huang, Wentao Liu, and Dahua Lin. Person search in videos with one portrait through visual and temporal links. In *Proc. ECCV*, 2018. 2

[7] Vicky Kalogeiton and Andrew Zisserman. Constrained video face clustering using 1nn relations. In *Proc. BMVC*, 2020. 8

[8] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 2, 3

[9] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008. 3

[10] Arsha Nagrani and Andrew Zisserman. From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. In *Proc. BMVC*, 2017. 3, 7

[11] Anindya Roy, Camille Guinaudeau, Hervé Bredin, and Claude Barras. Tvd: a reproducible and multiply aligned tv series dataset. In *LREC*, 2014. 7

[12] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proc. CVPR*, 2019. 8

[13] Kate Sim, Andrew Brown, and Amelia Hassoun. Thinking through and writing about research ethics beyond "broader impact". *CoRR*, 2021. 1

[14] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *Proc. ICCV*, 2019. 8

[15] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Learning to classify images without labels. *arXiv preprint arXiv:2005.12320*, 2020. 3

[16] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Utterance-level aggregation for speaker recognition in the wild. In *Proc. ICASSP*, 2019. 3