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

Andrew Brown[1], Vicky Kalogeiton[1,2], and Andrew Zisserman[1]
[1]University of Oxford, [2]École Polytechnique
{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

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. *VPCD* Details

Here, we give additional details on the annotation (Section 2.1) and feature extraction (Section 2.2) 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 2.3).

2.1. 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.

2.2. 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.

2.3. *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 3).

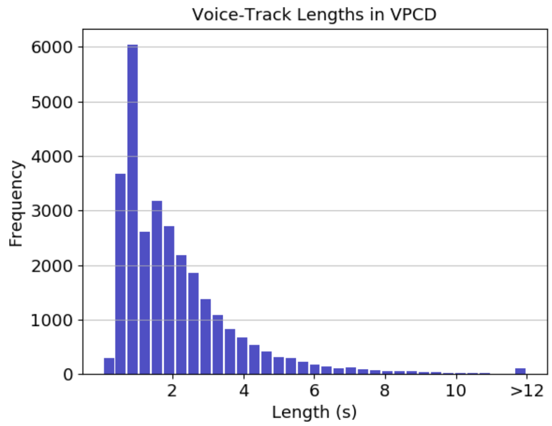


Figure 1: **Voice-track lengths in *VPCD*.** The distribution of all voice-track lengths in *VPCD*.

3. 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 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").

4. 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

	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 2.1). 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.

Character Precision and Recall (CP, CR).

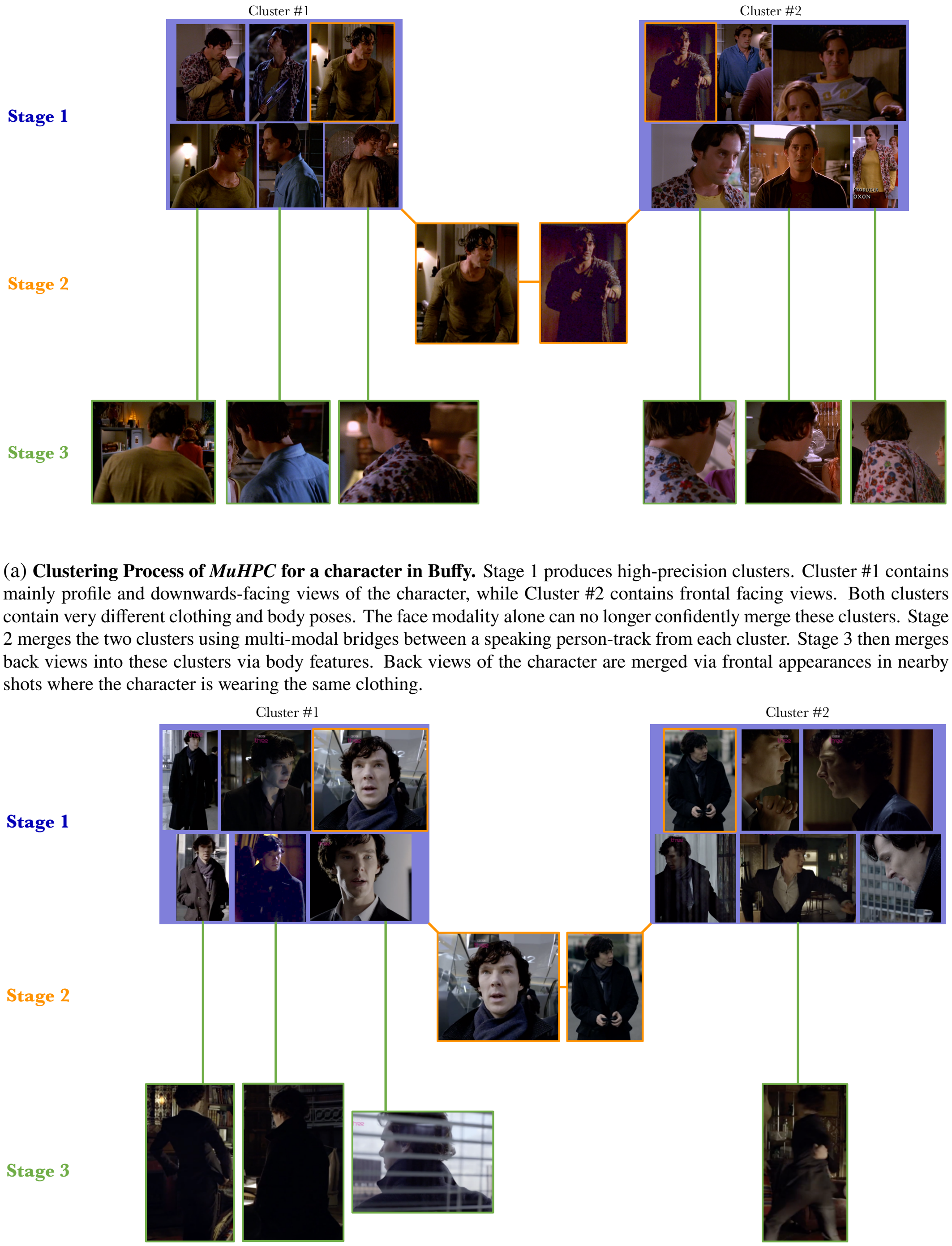
Weighted Clustering Purity (WCP). WCP weights the purity of a cluster by the number of samples belonging in it; to compute purity, each cluster c containing n_c elements is assigned to the class which is most frequent in the cluster. WCP is highest at 1 when within each cluster, all samples are from 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, $\text{NMI}(Y, C) = 2\frac{I(Y;C)}{H(Y)+H(C)}$, 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.

5. 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.

6. 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.

7. 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 7.1). 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 7.3).

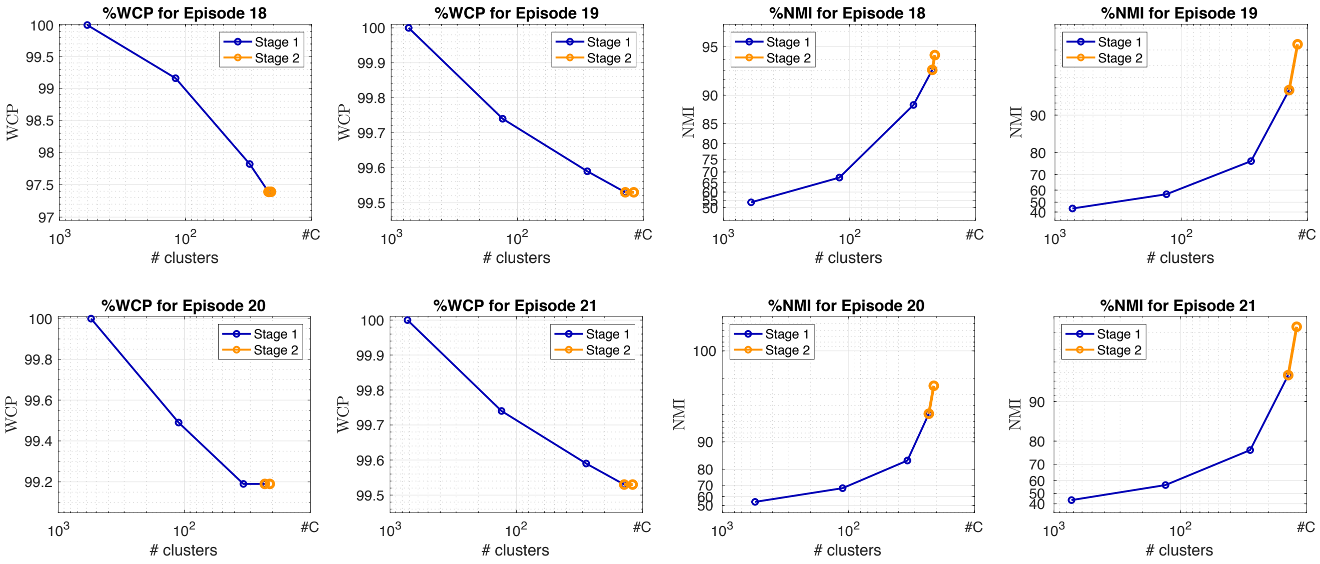


Figure 3: **Stage 1 and Stage 2 Person-Clustering results from the program set, Friends.** %WCP and %NMI for episodes of Friends from *VPCD*, for the Automatic Termination protocol (AT). The blue line illustrates the results after Stage 1, while the orange one illustrates the results after Stage 2, *i.e.*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

7.1. 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.

7.2. 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.

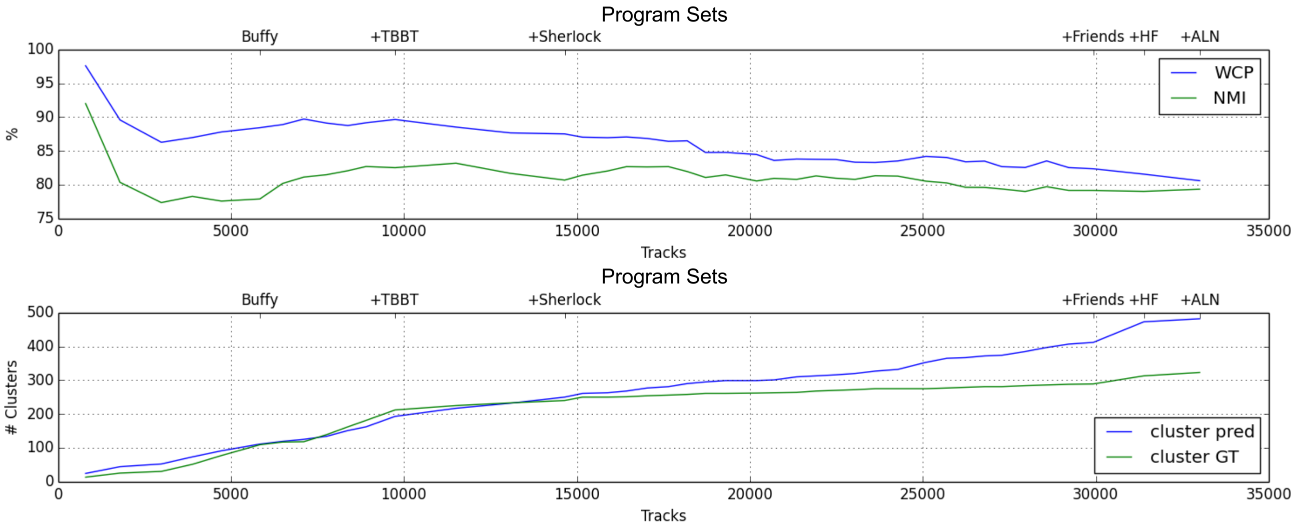


Figure 4: **Person-Clustering Results when clustering multiple program sets simultaneously.** Incrementally, more and more tracks are considered by adding different program sets together. There are discrete data points for each time the tracks from an additional episode or movie are added. Each data 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.

7.3. 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].

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, whereas *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.

8. 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.

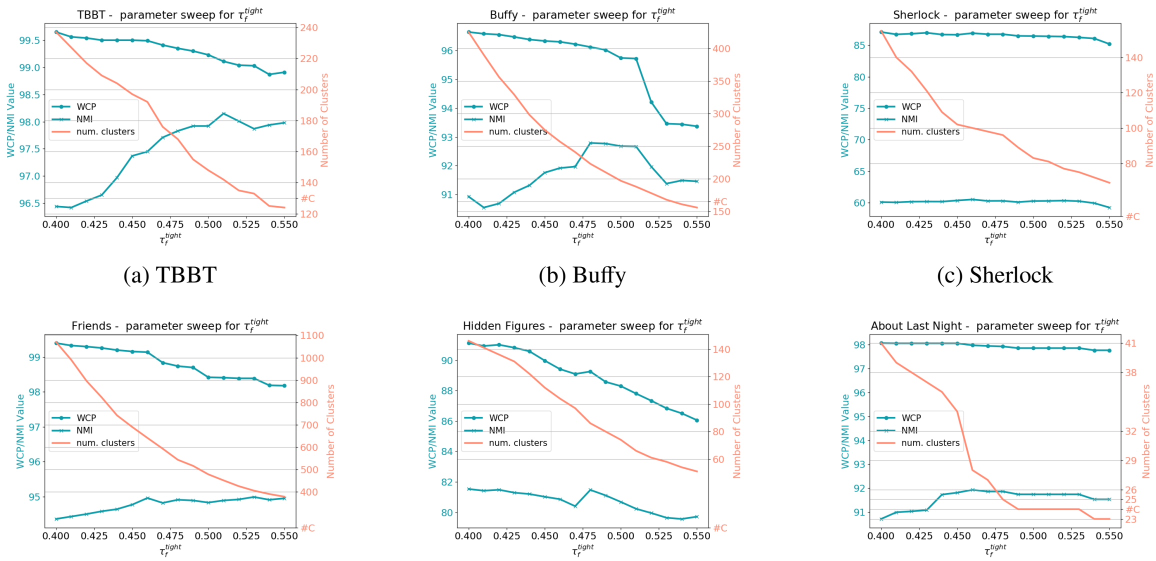
First, we analyse the AT protocol results. The goal of the AT protocol is to automatically terminate clustering and assess the quality of the resulting clusters. This is a realistic protocol for videos in-the-wild where the number of characters is unknown. Here, the number of predicted clusters, $\#C_p$, 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 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.

9. 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



(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.

3.2 in main manuscript).

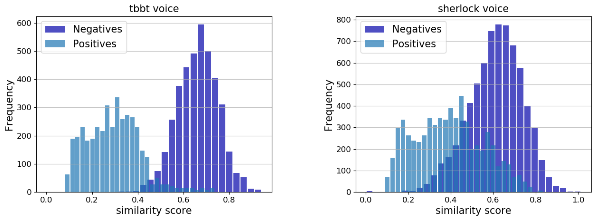
9.1. Nearest Neighbor Distance Threshold

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 8 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*.**

9.2. 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. 2