# Modular Memorability: Tiered Representations for Video Memorability Prediction – Supplemental

Théo Dumont
Mines Paris, PSL – Research University
Paris, France
dumont.theo@protonmail.com

Juan Segundo Hevia
Memorable AI
Boston, USA
juan.hevia@memorable.io

Camilo L. Fosco*
Memorable AI
Boston, USA
camilo@memorable.io

## A. Appendix

### A.1. Additional ablation studies

We report here the results of the following ablation studies: leave-one-out ablation of the raw perception module alone (Tab. 1), leave-one-out ablation of the raw perception module within the complete M3-S model (Tab. 2), and a comparative ablation study for the M3-S and M3 (no similarity module) models (Tab. 3). The leave-one-out ablations follow the method depicted below in Figure 1.

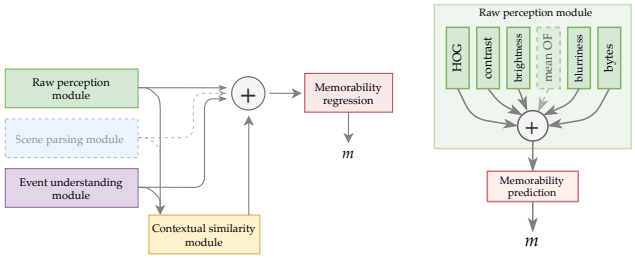



Figure 1. **Diagram of the ablation study. Left:** M3-S model with ablation of the scene understanding module. The ablation studies for the M3-S model consist in removing one module and evaluating the resulting model after training (leave-one-out). **Right:** Ablation study for the raw perception module (leave-one-out).

We find that on the complete M3-S model, the most important raw perception feature is the mean optical flow, which can be explained by the fact that the information it encompasses is not contained in any of the other modules. Moreover, we observe that the ablation study for the raw perception module alone does not necessarily gives insight into the results of the raw ablation on the complete M3-S model: although removing the blurriness feature seems to increase the raw perception module when used alone, this increase does not occur when removing the feature from the complete model.

*Corresponding author.

Table 1. **Ablation study of the raw perception module** on the Memento10k dataset (leave-one-out ablation).

| | | Memento10k [6] | |
|---|---|---|---|
| **Model** | **Features** | $\rho \uparrow$ | **MSE** $\downarrow$ |
| All features | 15 | 0.3989 | 0.00950 |
| Without blurriness | 14 | **0.4016** | **0.00947** |
| Without brightness | 14 | 0.3972 | 0.00955 |
| Without contrast | 14 | 0.3893 | 0.00960 |
| Without size | 14 | 0.3862 | 0.00960 |
| Without meanOF | 14 | 0.3559 | 0.00988 |
| Without HOG | 5 | 0.3077 | 0.01032 |

Table 2. **Ablation study of the raw perception module within the M3-S model** on the Memento10k dataset (leave-one-out ablation). * By unfrozen we mean that each module is trained; not only the feature aggregation MLPs.

| | | Memento10k [6] | |
|---|---|---|---|
| **Model** | **Features** | $\rho \uparrow$ | **MSE** $\downarrow$ |
| Full model, unfrozen* | 3,635 | **0.6699** | **0.00621** |
| Full model | 3,635 | 0.6349 | 0.00667 |
| Without brightness | 3,634 | 0.6324 | 0.006748 |
| Without contrast | 3,634 | 0.6317 | 0.006741 |
| Without blurriness | 3,634 | 0.6310 | 0.006748 |
| Without HOG | 3,625 | 0.6307 | 0.006733 |
| Without size | 3,634 | 0.6304 | 0.006754 |
| Without meanOF | 3,634 | 0.6247 | 0.006839 |

### A.2. Similarity module: additional results

**Similarity measures.**
We explored a large number of similarity measures to find the most relevant for video memorability prediction. We compute each similarity measure on the Memento10k dataset, and on both HRNet and CSN feature vectors after

Table 3. **Comparative ablation studies** on the Memento10k dataset. We report performance on M3 and M3-S checkpoints with frozen backbones — only the feature aggregation MLPs are trained.

| | Memento10k [6] | | | | Modules used | | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | $\rho \uparrow$ | **MSE** $\downarrow$ | **Features** | **Parameters** | Raw | Scene | Event | Sim. |
| M3-S | **0.6349** | **0.00667** | 3,635 | $1.89.10^6$ | ✓ | ✓ | ✓ | ✓ |
| M3 | <u>0.6303</u> | <u>0.00674</u> | 2,783 | $1.46.10^6$ | ✓ | ✓ | ✓ | |
| M3-S without raw | 0.6249 | 0.00685 | 3,620 | $1.89.10^6$ | | ✓ | ✓ | ✓ |
| M3 without raw | 0.6215 | 0.00686 | 2,768 | $1.45.10^6$ | | ✓ | ✓ | |
| M3-S without scene | 0.6139 | 0.00705 | 2,915 | $1.53.10^6$ | ✓ | | ✓ | ✓ |
| M3 without scene | 0.6136 | 0.00704 | 2,063 | $1.09.10^6$ | ✓ | | ✓ | |
| M3-S without event | 0.5692 | 0.00779 | 1,587 | $8.45.10^5$ | ✓ | ✓ | | ✓ |
| M3 without event | 0.5568 | 0.00789 | 735 | $4.09.10^5$ | ✓ | ✓ | | |

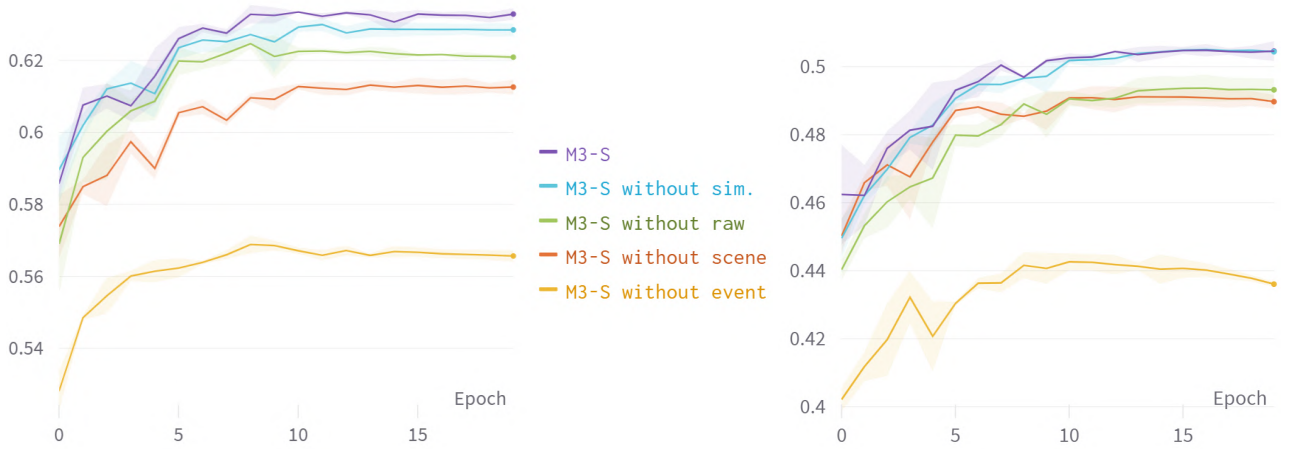



Figure 2. **Validation Spearman rank correlation scores** for the ablation runs on Memento10k **(left)** and VideoMem **(right)**. Models are trained for 20 epochs but tend to reach their maximum rank correlation around 10 epochs. We run each training test $N = 4$ times and report the average of the rank correlations (solid line) and the standard deviation (shaded area).

reduction to $d = 10$ components using PCA. Let us consider the semantic feature vector $F_0$ of a video clip. We want to quantify the distinctiveness — or equivalently its similarity — of feature $F_0$ comparatively to the training set.

As we are dealing with high-dimensional vectors, we avoid using the euclidean distance in a naive way, as it is known to behave badly in high-dimensional spaces [1], especially when using it with a view of quantifying distinctiveness. Aggarwal *et al.* [1] propose an alternative to this approach, which is to use a fractional distance with parameter $f \in (0, 1)$, defined by

$$\text{dist}_d^f(x, y) = \sum_{i=1}^{d} \left[ (x_i - y_i)^f \right]^{1/f}.$$

The first similarity measure we explore is therefore computing the mean of the fractional distance of $F_0$ with every training vector, using $f = 0.5$ **(1.a)**. As this measure is not very representative of the diversity of training samples, we also make use of the action class labels provided by the Memento10k dataset [6] by introducing for each of the 551 distinct action labels $(C_i)_i$ what we call a *prototype vector*, defined as the average vector of all train videos of the dataset that are of class $C_i$. We then compute the fractional distance from $F_0$ to each of the 551 prototype vectors and use this as our similarity features **(1.b)**. We also take the opportunity to compute this metric using the euclidean distance for the sake of completeness **(2)**. Another alternative to the euclidean distance is the cosine similarity. We therefore compute the cosine distances from $F_0$ to every element of the training set, and we either compute the mean of these distances **(3.a)** or introduce a threshold $T \in (0, 1)$ and consider the proportion of videos whose cosine similarity with $F_0$ is greater than $T$. We use three values of $T$ (0.4, 0.5 and 0.7) and stack the resulting values **(3.b)**. Then, we perform Kernel Density Estimation (KDE) on the training set and estimate the negative log likelihood of $F_0$ under this distribution **(4.a)**, which is the approach taken by

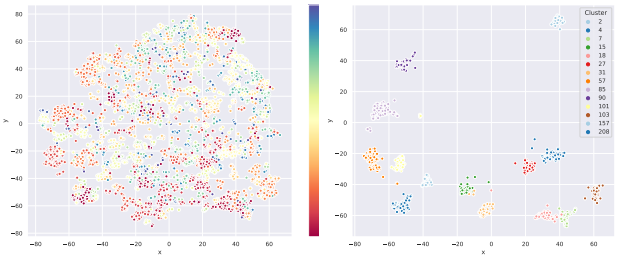
Bylinskii [3]. Still with the idea to exploit the diversity of the training set, we also perform this KDE under the distribution of each class-prototype, *i.e*. the set of videos that have the same action label **(4.b)**. Some classes do not have enough elements to produce a reliable density estimation, so we chose to only keep the 31 classes with the most elements in them (or equivalently the classes that contain at least 150 elements). Instead of using the already available classes of Memento10k and in order to gain generalizability, we also perform $k$-means on the feature space to produce 10 prototypes and perform the same similarity measure as before **(4.c)**. We experimented $k$-means with 10, 20, and 100 clusters and found that using 10 clusters was the best performing choice for our study. Finally, we examine the DBSCAN clustering results, discussed in the previous sections of the paper **(5)**. Results of this study can be found in Tab. 4.

We trained a simple MLP on 15 epochs to predict memorability based on the similarity features only. We leverage most similarity metrics with multiple different techniques and we report the results in this table. The *Metric* refers to the similarity metric used; the *Applied on* column tells if we apply the method to the whole training set, or only on some *prototype vectors*, average vectors of all train videos of class $C_i$, where classes $C_i$ are either obtained using Memento10k action labels or using the $k$-means algorithm; the *Type* gives additional information on how we aggregate the metrics across the similarity sets (*e.g.* averaging or considering the proportion of scores under a certain threshold). Finally, we define the *contribution* $\Delta\rho$ of a similarity measure $s$ to the M3-S model as the difference between the Spearman score of the model using $s$ and the M3 model, with no similarity module. For more details on the results and the techniques used, see the supplement.

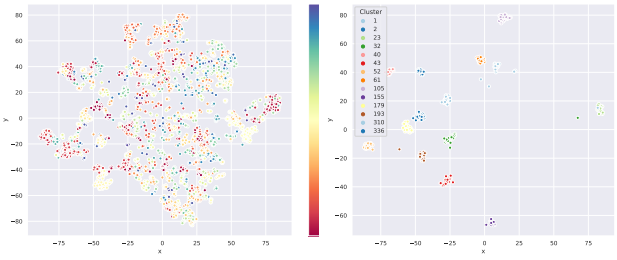
To evaluate the positive contribution of each similarity measure, we study the contribution $\Delta\rho$ to M3-S, being the difference between the Spearman score of the model using a measure and the M3 model, with no similarity module at all. It can be seen that using the mean euclidean or fractional distances to prototypes **(1.b, 2)** produces a very high Spearman Rank Correlation when used alone. However, this is not the metric that we want to optimize, since we mainly want the similarity module to improve the results of the M3-S model; and in term of $\Delta\rho$, these measures do not perform well. This could be explained by the fact that specifying the distances of a point $P$ to $p$ prototypes in an $n$-dimensional space roughly amounts to writing $p$ equations on the coordinates of $P$, which allow to locate the $P$ in a $(n-p+1)$-dimensional space. When $p$ grows closer to $n$, this amounts to pinpointing the location of the feature vector in its feature space, which is a very valuable information to predict memorability but is already contained in the original semantic feature vector. When adding these similarity measures to our M3-S model, we do not add any information and therefore see no improvement in term of Spearman RC.

From all the similarity measures explored, the DBSCAN results **(5)** and the mean fractional distance **(1.a)** performed best. We chose to keep the former for our M3-S model because of its interpretability.

**DBSCAN clustering interpretation.**
Plotting the video features as well as their cluster attribution in the t-SNE space allows to verify that DBSCAN creates clusters that are distinct in this representation space (Fig. 3). However, this characteristic is not sufficient for our study; we wish to verify that clustering produces groupings of video clips that allow us to discriminate them from each other, among other things by producing homogeneous clusters in terms of memorability. We thus represent the statistical characteristics of the memorability scores of the largest clusters in Fig. 4, and as the HRNet and CSN semantic features share similar properties, we focus on the latter for the rest of the study.



(a) HRNet features



(b) CSN features

Figure 3. **DBSCAN creates clusters that are distinct** in the t-SNE representation space for the two semantic features (HRNet and CSN). **Left:** Scatter plot of the entire set of clusters. **Right:** Scatter plot of the 15 largest clusters only. For the sake of clarity, we do not display the video clips that are not included in a DBSCAN cluster.

From this diagram, we obtain that each cluster has a different memorability range: some encompass video clips that are highly memorable (*e.g*. cluster 61) whereas others seem to group videos that are not (*e.g*. cluster 40). More importantly, we observe that the average memorability of

Table 4. **Predictive capacity of similarity measures alone** on the Memento10k dataset. We trained a simple MLP on 15 epochs to predict memorability based on the similarity features only. We explore most similarity methods in multiple different ways and we report the results in this table. The *Metric* refers to the similarity metric used; the *Applied on* column tells if we apply the method to the whole training set, or only on some *prototype vectors*, average vectors of all train videos of class $C_i$, where classes $C_i$ are either obtained using Memento10k action labels or using the $k$-means algorithm; the *Type* gives additional information on how we aggregate the metrics across the similarity sets (*e.g.* averaging, considering the proportion of scores under a certain threshold, or the frequency of the sample in the dataset given by a density estimation). We define the contribution $\Delta\rho$ of a similarity measure $s$ to the M3 model as the difference between the Spearman score of the model using $s$ and the M3 model, without similarity module, $\rho = 0.6291$ with standard hyperparameters.
*The total number of features of a similarity measure is the sum of that of HRNet and that of CSN. They are identical for all measures but DBSCAN, for which the number of clusters depends on the dataset used. This total number is therefore 699 for VideoMem and 852 for Memento10k. Please note that this corresponds to the length of the one-hot encoded cluster id, and that the feature vector is therefore very sparse (only one non-zero coordinate).

| | | | | | Predictive capacity | | Contrib. | Applicable to | |
|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **Applied on** | **Type** | **Ref.** | **Features*** | $\rho\uparrow$ | **MSE** $\downarrow$ | $\Delta\rho\uparrow$ | Memento10k | VideoMem |
| Fractional | training set | mean | (1.a) | $2 \times 1$ | 0.235 | 0.01091 | $+\underline{0.0029}$ | ✓ | ✓ |
| | proto. (labels) | mean | (1.b) | $2 \times 551$ | $\underline{0.491}$ | $\underline{0.00919}$ | $-0.0138$ | ✓ | |
| Euclidean | proto. (labels) | mean | (2) | $2 \times 551$ | **0.514** | **0.00852** | $-0.0021$ | ✓ | |
| Cosine | training set | mean | (3.a) | $2 \times 1$ | 0.151 | 0.01118 | $+0.0017$ | ✓ | ✓ |
| | training set | threshold | (3.b) | $2 \times 3$ | 0.321 | 0.01033 | $+0.0008$ | ✓ | ✓ |
| KDE | training set | density | (4.a) | $2 \times 1$ | 0.223 | 0.01090 | $+0.0021$ | ✓ | ✓ |
| | proto. (labels) | density | (4.b) | $2 \times 31$ | 0.449 | 0.00928 | $-0.0025$ | ✓ | |
| | proto. ($k$-means) | density | (4.c) | $2 \times 10$ | 0.409 | 0.00957 | $+0.0011$ | ✓ | ✓ |
| DBSCAN | training set | clustering | (5) | 699 / 852 | 0.340 | 0.01026 | $+$**0.0053** | ✓ | ✓ |

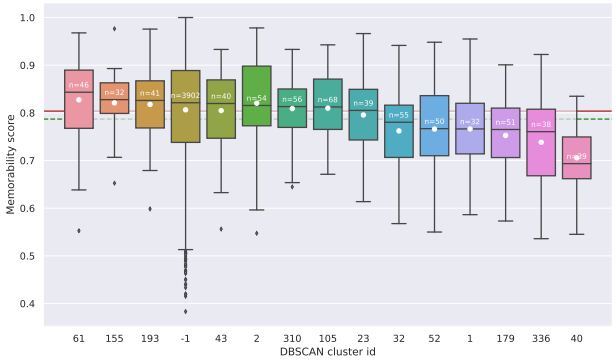
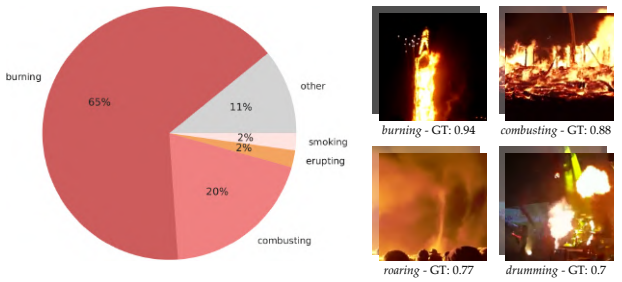



Figure 4. **Clusters learned have various ranges of memorability scores.** For each of the 15 largest DBSCAN clusters, we plot its quartiles, mean, outliers and number of observations `n`, as well as the average memorability score of the dataset (solid red line) and the average memorability score of the 15 largest clusters (dashed green line). Clusters are sorted by decreasing median value. The `-1` cluster corresponds to feature vectors that do not belong to a cluster. We show here the visualization that has been done on Memento10k using the DBSCAN clusters for the CSN semantic features.

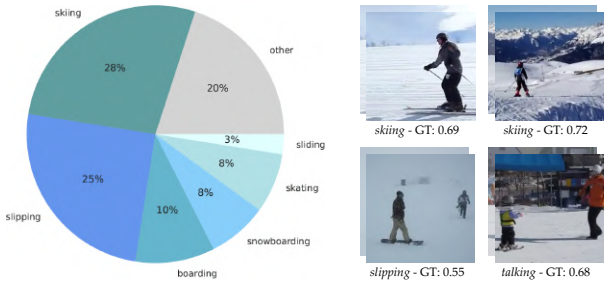
video clips belonging to one of the largest clusters is significantly lower than the average memorability on the complete dataset. As expected, this indicates that clips whose semantic content is very common tend to be less memorable than the others. Note that cluster `-1` is much bigger than all the other clusters combined, which means that it impacts a lot the average memorability of the 15 largest clusters; without it, the score would be even lower. To further investigate the relevance of our clusters, we analyse the semantics of the video clips they contain. We make use of Memento10k's action labels and report their distribution within clusters 61 and 40 (Fig. 5). We notice that the semantics and concepts of the video clips within the two clusters are indeed similar.

## A.3. Implementation details

**Feature generation and parameters.**
*Similarity module.* The similarity module takes each semantic feature vector and performs a DBSCAN clustering on it. As DBSCAN cannot be used to predict test class labels, we only perform the clustering on the training set and we use the results to train a MLP that will be used to predict the class labels on the test set. Before doing so, we perform a PCA to only keep the first 10 components; we then perform a t-SNE (of perplexity `30.0`) on the training set to further reduce this number to 3. We finally perform a DBSCAN clustering on the training data, that produces a list of classes, and we use this list as a training set for our predictor MLP, that takes a 10-feature long vector and outputs a cluster id. We use a value of epsilon of `1.25` for HRNet features and of `0.9` for CSN features, and the default values of `sklearn.cluster.DBSCAN` for the rest of the




(a) Memorable DBSCAN cluster (id 61, average memorability 0.82).




(b) Non-memorable DBSCAN cluster (id 40, average memorability 0.70).

Figure 5. **Semantic relevance of DBSCAN clusters. Left:** Action labels distribution within the cluster. **Right:** Examples of video clips from the cluster. Each cluster corresponds to one or several concepts, *e.g. fire* (a) or *skiing* (b). Video clips falling in the "*other*" category still share the same semantic content than the rest of the cluster: in cluster (a), *drumming* is indeed the main concept of the last video clip but fire occupies the frame; in cluster (b), the core concept of the last clip is rather *skiing* than *talking*.

parameters. We then convert the cluster id into a one-hot vector and use it as an input to our M3-S model.

*Feature computation.* Each video is resized to $256 \times 256$ and has its values rescaled between 0 and 1 before the computation of the low-level or semantic features. The low-level features are computed on each frame of the video and averaged to produce a single score/vector by feature for each video. As HRNet is an image segmentation model, we compute its output on the first frame of each video. In contrast, CSN can take a video of any length in input, therefore we compute its output on the whole video.

**Model structure and training.**

*MLP.* All the MLP used in the experiments are composed of a succession of $L$ layers of the form

$$f_k(x) = \sigma_k(A_k x + b_k)\,,$$

for $k = 1, \ldots, L$, where $A_k \in \mathbb{R}^{d_k \times d_{k+1}}$, $b_k \in \mathbb{R}^{d_{k+1}}$ are the weights and bias of layer $k$, $\sigma_k$ its activation function, always Mish except for the last one which is a sigmoid. The input dimension $d_0 := F_{\text{raw}} + F_{\text{scene}} + F_{\text{event}} + F_{\text{sim}}$ is the number of features and the output dimension $d_L$ is equal to 1. To analyse the predictive capacity of raw descriptors in our paper, we choose $L = 2$ and $(d_0, d_1, d_2) = (d_0, 64, 1)$. For the M3-S experiments, we choose $L = 3$ and $(d_0, d_1, d_2, d_3) = (d_0, 512, 64, 1)$.

*Loss functions.* We tested several loss functions for the training of our best performing models: the classic MSE loss, with or without a positive penalization $p(m)$ for the tails of the memorability scores distribution, a Spearman Rank Correlation loss [2], and a linear combination of both MSE and Spearman loss. We predicted and observed empirically that the Spearman loss allowed to obtain a very good correlation score, at the cost of a good performance in term of MSE. The ranks of the samples were indeed being correctly predicted, while the predicted distribution was close to normal. In order to keep the best of both scenarios, we chose to use a linear combination of these two loss functions, starting from MSE at epoch 0 and ending with Spearman at epoch $N_{\text{ep}}$, with a progressive transition in-between. This increased the correlation score on VideoMem by around 0.008. On the Memento10k dataset, the performance gain was negligible and we chose to keep the MSE loss function with penalization. Finally, for the sake of simplicity, we used the classic MSE loss for all the ablation studies.

*Training procedure.* In Tab. 5 can be found the details and parameters of our training runs. Having pre-computed the features for the four modules, one training session took in average 2.5 minutes on a AWS Tesla V100-SXM2-16GB.

Table 5. **Parameters used for the training procedure** of the M3-S model on the Memento10k and VideoMem datasets. All the parameters are kept the same across the two training procedures ("–" symbol), except the loss function.

| | **Value** | |
|---|---|---|
| **Parameter** | Memento10k [6] | VideoMem [4] |
| Hidden channels | $[512, 64, 1]$ | – |
| Batch size | 32 | – |
| Learning rate | $10^{-3}$ | – |
| Scheduler | StepLR, $\gamma = 0.2$, step size $= 5$ | – |
| Epochs | 20 | – |
| Loss | MSE (tails) | MSE + Spearman RC |
| Weight decay | $10^{-5}$ | – |
| Optimizer | Adam | – |
| Normalizing raw | ✓ | – |
| Normalizing sim | ✓ | – |

## A.4. Reproducibility Statement

The Python code as well as the pretrained weights are available at https://github.com/tekal-ai/modular-memorability. Additionally, in order to slightly mitigate the impact of variability in the training process, we always performed $N \geq 4$ training experiments and

reported the mean $\bar{\rho}$ of the Spearman rank correlation coefficients $\rho$ in the tables.

### A.5. CO2 Emission Related to Experiments

Experiments were conducted using Amazon Web Services in region us-east-1, which has a carbon efficiency of 0.37 kg$CO_2$eq/kWh. A cumulative of 175 hours of computation was performed on hardware of type Tesla V100-SXM2-16GB (TDP of 250W). Total emissions are estimated to be 16.19 kg$CO_2$eq of which 0 percents were directly offset by the cloud provider.

Estimations were conducted using the Machine Learning Impact calculator presented in [5].

## References


[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001. 2

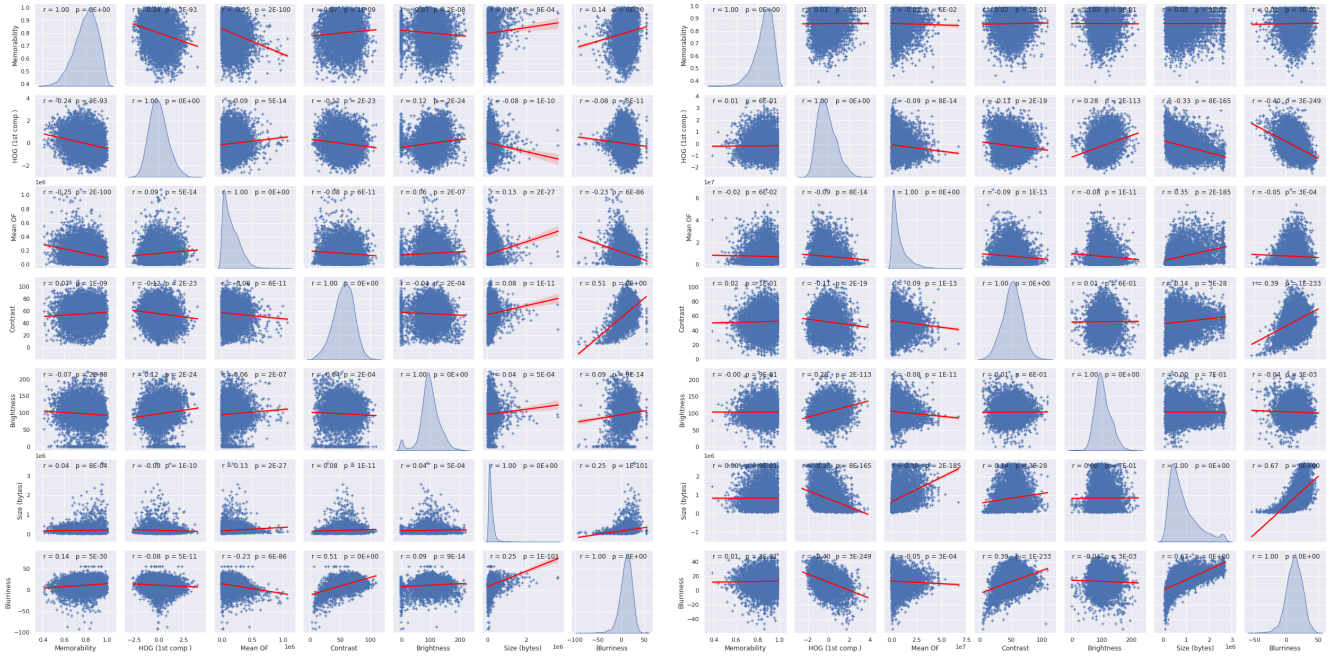
[2] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020. 5

[3] Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and extrinsic effects on image memorability. *Vision research*, 116:165–178, 2015. 3

[4] Romain Cohendet, Claire-Hélène Demarty, Ngoc QK Duong, and Martin Engilberge. Videomem: constructing, analyzing, predicting short-term and long-term video memorability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2531–2540, 2019. 5, 7

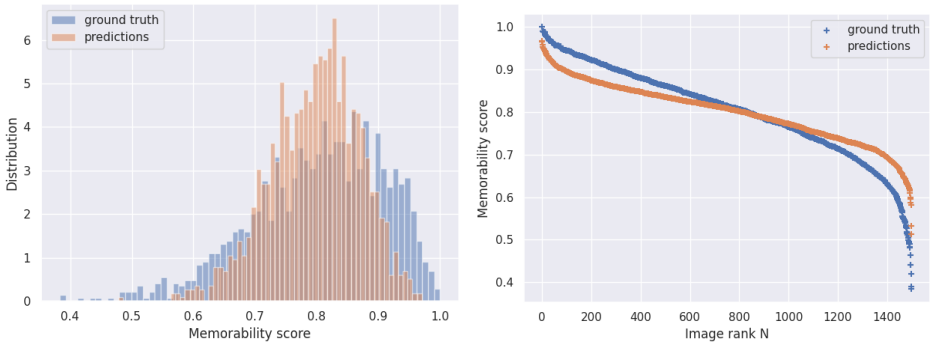
[5] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019. 6

[6] Anelise Newman, Camilo Fosco, Vincent Casser, Allen Lee, Barry McNamara, and Aude Oliva. Multimodal memorability: Modeling effects of semantics and decay on video memorability. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 223–240. Springer, 2020. 1, 2, 5, 7

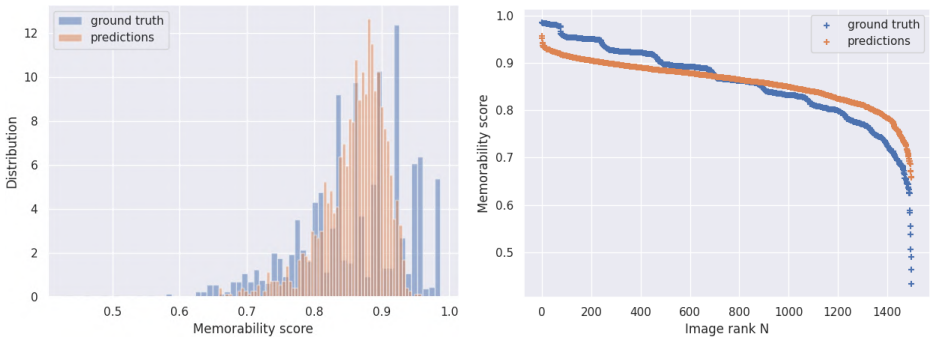



Memento10k [6] VideoMem [4]

Figure 6. **Pairwise relationships between the raw features and memorability** for Memento10k **(Left)** and VideoMem **(Right)**. We report the Pearson correlation coefficient $r$ as well as the $p$-value $p$ in each cell of the grid.




(a) Memento10k [6]




(b) VideoMem [4]

Figure 7. **Distributions of ground truth and predictions** on Memento10k **(a)** and VideoMem **(b)**. **Left:** Distribution of the ground truth and of the predicted scores of our M3-S model. **Right:** Ranking distribution for ground truth and predicted memorability scores. Videos are ranked by memorability scores and the ranks are plotted against the memorability scores.