A Methodology to Evaluate Strategies Predicting Rankings on Unseen Domains

Sébastien Piérard, Adrien Deliège, Anaïs Halin, Marc Van Droogenbroeck
Montefiore Institute, University of Liège, Belgium
{S.Pierard,Adrien.Deliege,Anais.Halin,M.VanDroogenbroeck}@uliege.be

***Abstract*—Frequently, multiple entities (methods, algorithms, procedures, solutions, etc.) can be developed for a common task and applied across various domains that differ in the distribution of scenarios encountered. For example, in computer vision, the input data provided to image analysis methods depend on the type of sensor used, its location, and the scene content. However, a crucial difficulty remains: can we predict which entities will perform best in a new domain based on assessments on known domains, without having to carry out new and costly evaluations? This paper presents an original methodology to address this question, in a leave-one-domain-out fashion, for various application-specific preferences. We illustrate its use with 30 strategies to predict the rankings of 40 entities (unsupervised background subtraction methods) on 53 domains (videos).**

***Index Terms*—performance, multi-domain, ranking prediction, evaluation methodology, Tile, background subtraction**

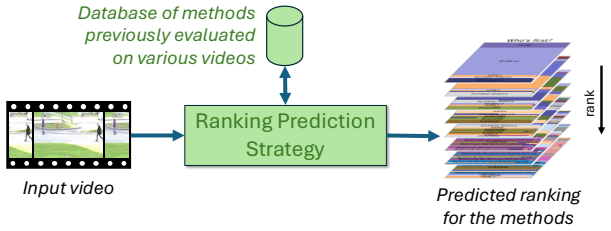


Fig. 1: In this paper, we explore the problem of predicting the rankings of computer vision methods (we take the particular case of background subtraction methods) on any new domain (video) based on a database storing the performances of these methods, previously evaluated on other domains (videos).

I. Introduction

Why do computer scientists rank entities (methods, algorithms, etc.)? Certainly, not just for picking a winner for a contest. The aim is rather to gain some knowledge that can help to select the most promising entity for a particular application. Predicting a ranking is a topic that has rarely been studied. As depicted in Fig. 1, we explore the case of a computer vision task and present an original methodology for investigating this theme, based on recent theory [1] and tools [2], [3] that are briefly recalled.

Background subtraction (BGS) is a computer-vision task for which the established rankings drastically differ from a video to another. Undoubtedly, this is due, on the one hand, to the variety of principles supporting these methods and, on the other hand, to the wide range of potential applications, and so to the wide variety of videos on which these methods can be applied. As a result, practitioners are frequently confronted with the selection of the most appropriate method for new video sequences, new camera placements, or video streams with characteristics changing unpredictably over time.

A first path consists in predicting the same ranking for all applicative domains, regardless of their characteristics. This is expected to be suboptimal, but it is worth trying such strategies for the sake of simplicity.

A second path consists in predicting the rankings based on the knowledge the practitioner has about his/her applicative domain. For example, he/she can specify a category to which all the inputs belong. The dataset *CDnet 2012* [4] defined 6 categories: *baseline*, *dynamic background*, *camera jitter*, *intermittent object motion*, *shadow*, and *thermal*, later complemented in *CDnet 2014* [5] by *bad weather*, *low framerate*, *night videos*, *PTZ*, and *turbulence*.

A third path consists in analyzing the input and predicting the ranking based on the perceived domain characteristics. Practitioners can collect representative images and analyze them to characterize their applicative domains as statistical distributions. For example, Halin *et al.* [6] have demonstrated the feasibility of characterizing domains, based on images, in a physically interpretable way with simulation-based inference [7] techniques. Alternatively, one could also classify pixels (*e.g.*, with semantic segmentation) and characterize the domain by the predicted distribution of classes.

In fact, our BGS example is just a particular case of a general and common problem that consists in predicting the rankings of entities for unseen domains based on performance evaluations previously performed for other domains, without having to carry out any new and costly evaluation. We go beyond just predicting the "best" entity as a user could be constrained to any subset of entities. To evaluate a predictive strategy, we compute the probability that, for any randomly chosen pair of entities (ϵ_a, ϵ_b), ϵ_a is predicted as worse or better than ϵ_b when it is actually the case.

Currently, scientists are not aware of how well simple and intuitive strategies perform. The ability to exploit existing rankings is all a blur. Our objective is to provide methodological elements and rigorous tools to clarify this. An originality of this work is that we do not rely on a small set of scores, but on a large family of scores, all of which have the required properties to induce meaningful rankings covering a diversity of application-specific preferences.

Our two main contributions are the following. (1) First, we establish a new and original methodology to evaluate and compare strategies predicting rankings on unseen domains. It leverages the recently introduced theoretical foundations for performance-based ranking [1] as well as a recently described tool (*i.e.*, the Tile [2], [3]). This framework allows us to cover an infinite and diversified family of meaningful performance-based rankings and to see how strategies perform according to application-specific preferences. (2) Second, we show a new usage for the Tile, originally introduced as a graphical tool to analyze and compare performances of two-class classifiers. In this work, the Tile appears to be the appropriate tool to depict the performance of strategies predicting rankings of two-class classifiers on unseen domains, and to compare them.

II. Related Work

A. Background Subtraction

Most of the literature on BGS is about the methods [8]–[10], with some intended to be all-purpose (*i.e.*, universal) while others are developed for specific application domains [11]. Some consider constraints such as limited resources (time, memory, etc.) [12], [13], and some tackle multimodality [14].

A small part of the literature also concerns datasets, initially used to compare the BGS methods, then leveraged to develop supervised methods [8]. The best-known datasets are *CDnet 2012* [4] and its extension *CDnet 2014* [5], which are integrated into the changedetection.net platform.

Eventually, a part of the literature focuses on performance analysis. BGS is typically identified to a pixel-based two-class classification, where the negative and positive classes correspond to the background and the foreground (in the following, we denote a true negative, false positive, false negative, and true positive by tn, fp, fn, and tp). Jodoin *et al.* [15] and Piérard *et al.* [16] determined which performances are achievable by combining existing BGS methods. Braham *et al.* [17] introduced a generic technique to improve BGS methods performances. Piérard *et al.* [18] studied the limits of achievable performances with simple methods (pixelwise color comparisons), and how they vary with some properties (*e.g.*, amounts of noise and shadows) of the videos analyzed. Piérard *et al.* [19] highlighted the pitfalls of averaging values of scores measured on several videos, and provided a technique for summarizing the corresponding performances into a single interpretable one.

B. Performance-Based Ranking

Theoretical foundations for performance-based rankings, grounded in probability and order theories, have recently been introduced by Piérard *et al.* [1] through a rigorous axiomatic framework[1]. These axioms are guardrails to guarantee meaningful rankings while leaving the flexibility to adjust the rankings *w.r.t.* application-specific preferences.

[1]Their 1st axiom states that any performance-based ranking should be derived from a preorder on performances, which ensures the stability of the rankings *w.r.t.* insertions and deletions of ranked entities. Their 2nd axiom gives compatibility conditions between the preorders and the considered task, modeled by a random variable called *satisfaction*. Their 3rd axiom gives compatibility conditions between the preorders and known properties about the evaluation (*i.e.*, the mapping of the entities to their performances).

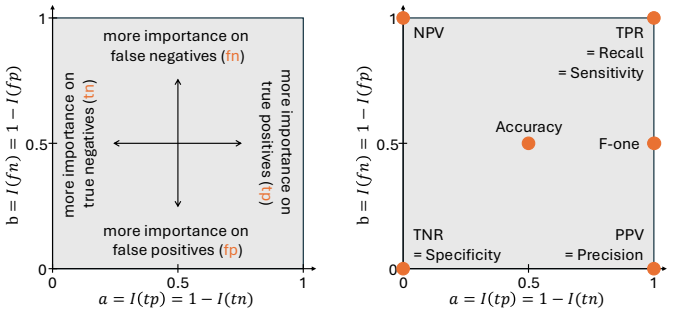


Fig. 2: Two equivalent readings of the Tile: a map of application-specific importances (left) and a map of scores to induce meaningful performance-based rankings (right).

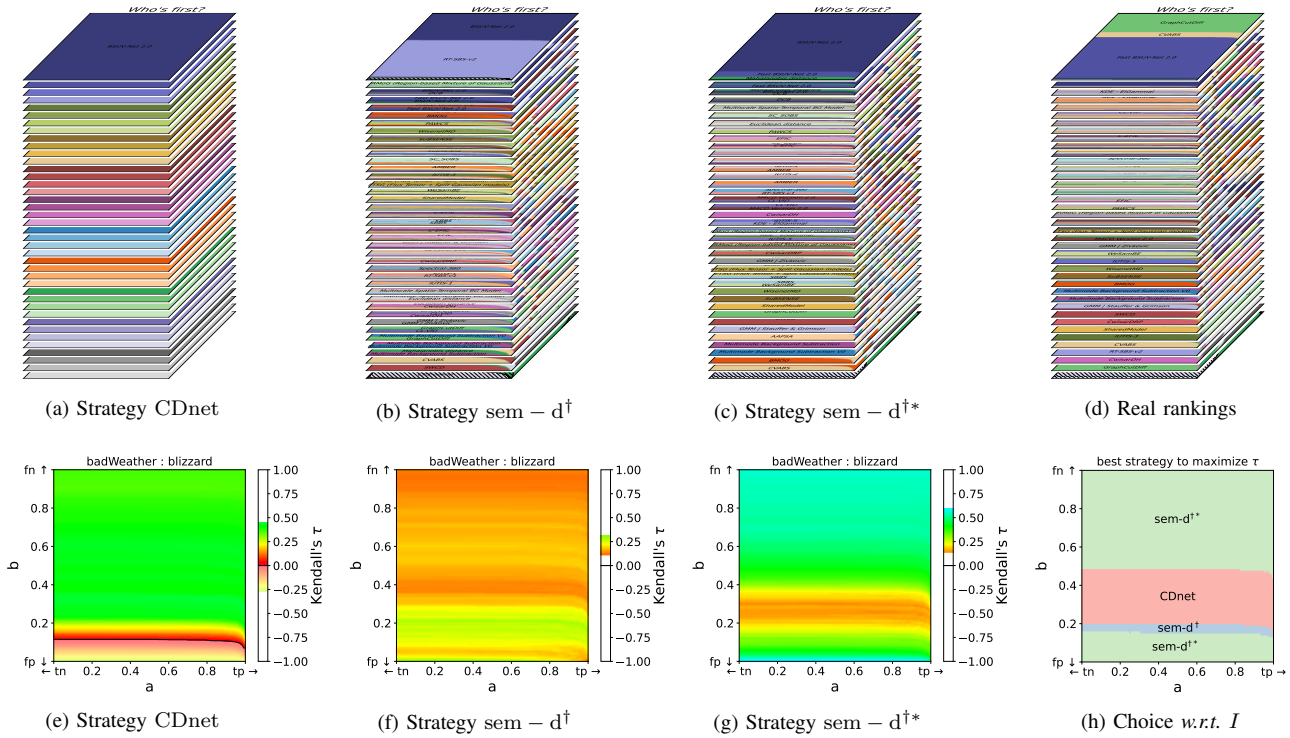
When particularized to two-class classification (*e.g.*, between background and foreground, or between negatives and positives), it has been shown in [1]–[3] that one can fine-tune the relative importance given to the two types of correct classifications, as well as to the two types of erroneous classifications, by inducing the performance ordering from a parametric score R_I called *ranking score*:

$$R_I : P \mapsto R_I(P) = \frac{\sum_{\omega \in \{tn,tp\}} I(\omega) P(\{\omega\})}{\sum_{\omega \in \{tn,fp,fn,tp\}} I(\omega) P(\{\omega\})}, \quad (1)$$

where P denotes the performance, and I the importance, *i.e.*, the application-specific preferences. Ranking scores sharing the same values for $a = \frac{I(tp)}{I(tn)+I(tp)}$ and $b = \frac{I(fn)}{I(fp)+I(fn)}$ lead to the same rankings [2]. So, the diversity of rankings can be shown on the $(a, b) \in [0, 1]^2$ square, coined as the Tile [2], [3]. Interestingly, the family of ranking scores contains scores commonly used in the BGS community (see Fig. 2).

C. Ranking Prediction

Predicting a ranking of entities, sometimes called "learning to rank" in machine learning, is a task used for instance in recommendation systems, or action quality assessment (*e.g.*, in sports). The strategies generally fall into three categories: *pointwise*, where strategies predict the performance score of each entity individually, and derive a ranking from these scores [20], [21]; *pairwise*, where strategies predict the relative ordering between entity pairs, and derive a ranking from pairwise orderings [22]; and *listwise*, where strategies predict directly the entire ranked list [23]. Pointwise approaches are simple and efficient but ignore the relative nature of ranking tasks. Pairwise strategies better capture it and are widely used, though they struggle with global ranking consistency. Listwise approaches, by directly optimizing ranking scores, often achieve superior performance, but are typically more complex and computationally demanding [24], [25]. Compared to that literature, we do not aim to learn a model that ranks entities based on domain (videos) features; instead, we leverage pre-computed rankings across multiple domains to infer the ranking on a previously unseen one.



(a) Strategy CDnet (b) Strategy $\mathrm{sem}-\mathrm{d}^{\dagger}$ (c) Strategy $\mathrm{sem}-\mathrm{d}^{\dagger *}$ (d) Real rankings

(e) Strategy CDnet (f) Strategy $\mathrm{sem}-\mathrm{d}^{\dagger}$ (g) Strategy $\mathrm{sem}-\mathrm{d}^{\dagger *}$ (h) Choice *w.r.t.* I

Fig. 3: Illustration of our methodology for the evaluation and comparison of 3 strategies to predict the rankings of 40 BGS methods on the video "bad weather: blizzard". We use Tiles [2] to cover the different application-specific preferences (a, b). The upper row shows: (a) the predicted ranking based just on the global ranking given on changedetection.net (strategy CDnet), (b) the one based on the performance measured on another semantically close video (strategy $\mathrm{sem}-\mathrm{d}^{\dagger}$), (c) the one based on the performance measured on another semantically close video in the same category (strategy $\mathrm{sem}-\mathrm{d}^{\dagger *}$), and (d) the actual ranking we would like to predict (our ground truth). These "mille-feuilles" are stackings of entity Tiles [3]: the k-th layer shows the methods ranked k-th, the worst methods being at the base of the mille-feuille, and the best ones on its top. The lower row shows the correlation Tiles [3] between these rankings: (e) the correlation between (a) and (d), (f) the correlation between (b) and (d), and (g) the correlation between (c) and (d). The Tile (h) shows which strategy gives the best correlation. This methodology is applied in Sec. IV to compare many more strategies on a diversified set of 53 videos.

III. Methodology

We now describe our new methodology to evaluate and compare strategies predicting the rankings of methods in unseen domains. We focus on strategies able to predict a large family of meaningful rankings: all those induced by ranking scores [1] (*cf.* Eq. 1). For problems comparable to two-class classification, these rankings can be mapped on the Tile [2], [3], which leads to a *mille-feuille*[2], as shown in Fig. 3.

We propose to evaluate strategies, for any given domain, with correlation Tiles [3] giving the rank-correlation between the predicted and ground-truth rankings, *w.r.t.* the various application-specific preferences (specified by I, or a and b). We choose Kendall's τ as this score is coherent with the performance-based ranking framework[3]. Moreover, Kendall's τ can be nicely interpreted as the probability of observing, in two given rankings, the same relative order between two randomly chosen methods, this probability being linearly mapped into the $[-1, +1]$ interval (see Fig. 4).

Considering several domains can be done in a *leave-one-domain-out* fashion, predicting the ranking for each domain using information from the others. The expected performance of the strategy is given by the *mean correlation Tile*, whereas the worst-case is given by the *minimum correlation Tile*. This will be illustrated in Sec. IV.

Two natural baselines can be defined when the ground-truth rankings are known for several domains. (1) The mean of τ over all domain pairs estimates the expected agreement when generalizing a ranking from a randomly chosen reference domain to another random domain. (2) The minimum of τ across all domain pairs captures the worst-case agreement for any fixed domain and arbitrary reference.

To compare various strategies, it is essential to bear in mind that the best strategy might depend on the application-specific preferences I. This can be mapped on the Tile.

[2]*Mille-feuille* pastries are composed of several layers, with some made with puff pastry, resembling to a stacking of multiple sheets.

[3]It turns out that Kendall's τ is itself a ranking score for ranking problems when pairs of miss-ordered and well-ordered methods are assigned a satisfaction of -1 and +1, respectively. See the appendix of [1] for more information.

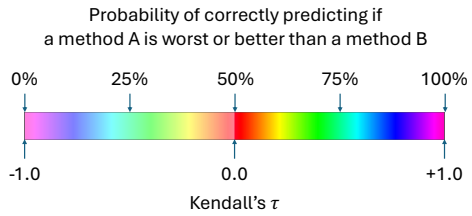


Fig. 4: Interpretation of the rank correlation τ (Kendall). Note the color code for displaying the value of τ, as is used in other figures.

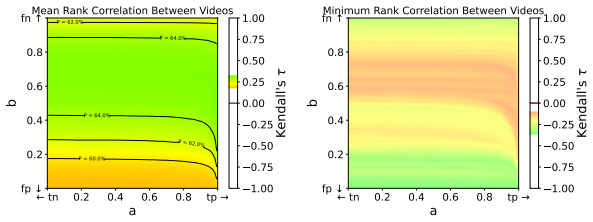


Fig. 5: The two natural baselines for ranking prediction strategies (as defined in our methodology) in the particular case of 40 unsupervised BGS algorithms ranked on 53 videos.

IV. Application

We now apply our methodology to the problem of predicting the performance-based rankings of 40 unsupervised BGS methods on the 53 videos of *CDnet 2014*. As said, we use only rankings, no implementation of these methods, ground-truth segmentation masks, nor any inference and evaluation step. In particular, the priors and the rates of predictions for the background and foreground are unknown. We compare strategies that have no knowledge about the input video (their expected distribution put aside) and strategies that leverage some domain characterization: a category (denoted by *), a distribution of semantic classes[4] (denoted by †), or both.

The information known about the test video (*i.e.*, for which we must predict the ranking) is its two-fold domain characterization: (1) its category in *CDnet 2014* and (2) an estimate of the statistical distribution of the pixel-based semantic labels. The information known for the others is three-fold: (1) its category, (2) the join distribution of the pixel-based semantic labels, ground-truth class, and estimated classes, and (3) the performance-based ranking of the 40 BGS methods. All distributions are for the publicly available subset of pixels annotated as background or foreground in *CDnet 2014*.

A. Baselines for Ranking Prediction Strategies

The two natural baselines defined in Sec. III are shown in Fig. 5. On the left-hand side, we see that there is only a little average correlation between rankings from one video to another. The mean value for τ around 0.25 means that there is about 62.5% chance of correctly ordering two methods if we refer to a predetermined ranking for any randomly chosen video. Moreover, we see that the difficulty of predicting rankings is not uniform: it seems to be a more difficult problem when $I(fn)$ is low *w.r.t.* $I(fp)$ (bottom of the Tile). On the right-hand side, we observe that, in the worst case, regardless of what the application-specific preferences I are, there can be a negative correlation between the rankings on two videos. All things considered, the problem of predicting the performance-based rankings of BGS methods is a difficult problem.

[4]The semantic labels are used only to cluster the pixels. Thus, we can rely on predicted (not perfect) segmentations. We selected a model from MMSegmentation [26] based on two criteria: semantic diversity and expected performance. For the former, we limited the choice to ADE20K [27] models and COCO-Stuff [28] models. Based on performance insights from *entity Tiles* [3], we selected Mask2Former [29] trained on ADE20K.

B. First Experiment

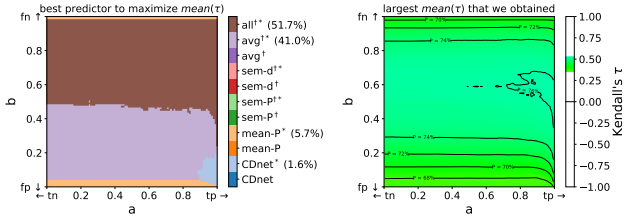
The aim of the 1st experiment is to compare a variety of 11 very intuitive strategies. [CDnet] is the ranking given on changedetection.net[5] at an arbitrarily chosen moment[6]. It is not sensitive to application-specific preferences. [mean-P] is the ranking based on the summarization [19] of the performances determined on the 52 other videos. [sem-P†] is the ranking based on the performance predicted from, on the one hand, the distribution of semantic labels in the input video and, on the other hand, the probabilities of false/true background/foreground conditionally to the semantic label for the other videos. [sem-d†] is the ranking known for the closest video in terms of semantic characterization (Bhattacharyya distance between distributions of semantic labels). [avg†] is the ranking based on the summarization [19] of the performances predicted for CDnet[7], mean-P, sem-P†, and sem-d†. [...*]: each of these five strategies comes with a variant that is category-specific and that is denoted by *. [all†] is the ranking based on the summarization [19] of the performances predicted for avg† and avg$^{\dagger *}$.

The results are shown in Fig. 6. We see that, when we want to maximize $mean(\tau)$, it is clearly useful to consider hybrid strategies, choosing one of the basic strategies according to the application-specific preferences: none of the 11 strategies tested in this 1st experiment is the best in more than 51.7% of the Tile. Moreover, we see that for the vast part of the Tile, it is best to opt for strategies based on the summarization of performances predicted in various ways. The maximum $mean(\tau)$ obtained is relatively independent of application-specific preferences (the Tile is all in green shades). When we want to maximize $min(\tau)$, the strategies to select are, over the majority of the Tile, strategies that exploit knowledge of the category to which the input video belongs.

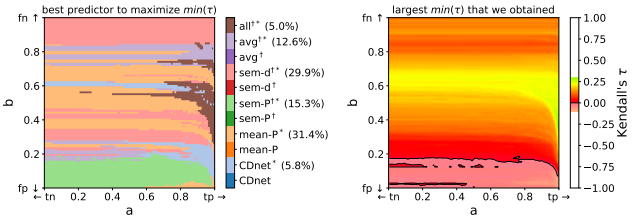
[5]This is our sole strategy that is not leave-one-video-out, as all our input videos are involved in this ranking. The related results could be optimistic.

[6]Version of March, 11th 2025. This remark is needed because the overall ranking of changedetection.net does not satisfy the 1st axiom of [1]: it is not stable. The relative order between the 40 methods, can be influenced by the presence of other methods (*e.g.*, the supervised ones) on the platform, and could change if a method was added or removed from the platform.

[7]Because the summarization only accepts performances as input, and not real values, we need a solution to convert a value into a performance. For any value $v \in [l, u]$, we assign the performance P such that $P(\{tn\}) = P(\{tp\}) = \frac{1}{2}\frac{v-l}{u-l}$ and $P(\{fp\}) = P(\{fn\}) = \frac{1}{2}\frac{u-v}{u-l}$, as $R_I(P)$ is strictly monotonously increasing with v for all ranking scores R_I.

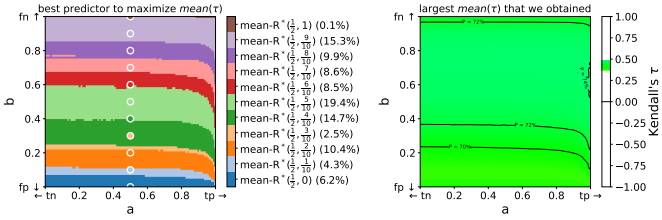


(a) Hybrid strategy maximizing $mean(\tau)$.

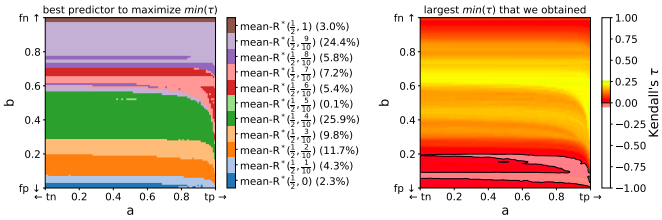


(b) Hybrid strategy maximizing $min(\tau)$.

Fig. 6: Results of our 1st experiment.



(a) Hybrid strategy maximizing $mean(\tau)$.



(b) Hybrid strategy maximizing $min(\tau)$.

Fig. 7: Results of our 2nd experiment.

C. Second Experiment

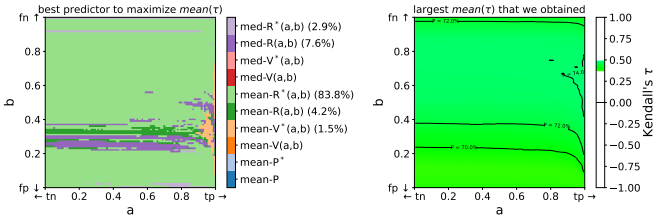
Now, we aim to establish something simpler, and easier to interpret, than the first experiment. We study the vertical behavior of the Tile through a parametric family of strategies. We focus on strategies that exploit the knowledge of the category, but put semantics aside. This experiment involves 11 strategies to predict rankings. [mean-R*$(\frac{1}{2}, x)$] is the category-specific weighted[8] arithmetic mean of the ranks induced by the canonical ranking scores $R_{I_{a,b}}$ with $a = \frac{1}{2}$ and $b = x$.

Results are shown in Fig. 7 for the discrete set of values $x \in \{0, \frac{1}{10}, \frac{2}{10} \dots 1\}$. When we want to maximize $mean(\tau)$, we see that all 11 strategies are useful. Moreover, the point $(a, b) = (\frac{1}{2}, x)$ lies in, or is not far from, the area of the Tile where the strategy mean-R*$(\frac{1}{2}, x)$ is the best (see the colored points). The average τ achievable with these strategies is close to the average τ obtained with the strategies of the first experiment. When we want to maximize $min(\tau)$, our conclusions are similar to those given for the maximization of $mean(\tau)$. This paves the way for an even more refined hybrid strategy, combining not only 11 strategies as in this experiment, but an infinite number of them (one for each a and b), which is done in the 3rd experiment.

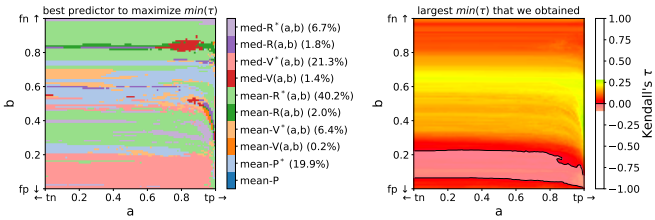
D. Third Experiment

Now, we aim to explore two questions: (1) is it better to compute average performances, average values, or average ranks? (2) is it more suitable to compute means or medians? This experiment involves 10 strategies. [mean-P] is the ranking based on the summarization [19] of the performances determined on the 52 other videos. [mean-V] is the weighted arithmetic mean of the values taken by the canonical ranking score, corresponding to the application-specific preferences. [mean-R] is the weighted arithmetic mean of the ranks induced by the canonical ranking score, corresponding to the application-specific preferences. [med-V] is similar to mean-V, except that we take the median instead of the mean. [med-R] is similar to mean-R, except that we take the median instead of the mean. [...*]: as in the 1st experiment, each of these five strategies comes with a variant that is category-specific and that is denoted by *.

[8]We weight videos as in changedetection.net: all categories have the same weight, and all videos in a given category have also the same weight. The video on which the rankings are predicted has a zero weight.



(a) Hybrid strategy maximizing $mean(\tau)$.



(b) Hybrid strategy maximizing $min(\tau)$.

Fig. 8: Results of our 3rd experiment.

The results of the 3rd experiment are displayed in Fig. 8. When we want to maximize $mean(\tau)$, in 83.8% of the Tile, mean-R* is the best strategy among the 10. When we want to maximize $min(\tau)$, for most of the Tile, it is better to take only strategies that exploit videos within the same category. The $mean(\tau)$ and $min(\tau)$ Tiles are very similar to the ones we obtained in our first two experiments.

V. Conclusion

This work introduces a methodology for comparing strategies for the prediction of rankings of computer vision methods (*e.g.*, BGS methods) on new domains (*e.g.*, videos), *w.r.t.* application-specific preferences, by exploiting the knowledge that can be found in publicly available multi-domain rankings (*e.g.*, changedetection.net). The methodology, as presented for a problem similar to the two-class classification, takes advantage of a recently introduced visualization tool called "Tile" [2], [3], which is based on a theoretical framework for performances [1]. In the case that we studied, we have shown that the performance-based rankings of computer vision methods on new domains is far from being an easy or solved problem. Because of its practical importance, we hope that this work will stimulate the community to investigate and develop more powerful strategies, and that our proposed methodology will help to establish their effectiveness.

Acknowledgment

S. Piérard, A. Deliège, and A. Halin are funded respectively by (1) the Service Public de Wallonie (SPW) Recherche (grant 8573, ReconnAIssance project), (2) ULiège (project DESTINA), and (3) the SPW EER, Wallonia, Belgium (grant 2010235, ARIAC by DIGITALWALLONIA4.AI).

References

[1] Sébastien Piérard, Anaïs Halin, Anthony Cioppa, Adrien Deliège, and Marc Van Droogenbroeck, "Foundations of the theory of performance-based ranking," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2025.

[2] Sébastien Piérard, Anaïs Halin, Anthony Cioppa, Adrien Deliège, and Marc Van Droogenbroeck, "The Tile: A 2D map of ranking scores for two-class classification," *arXiv*, vol. abs/2412.04309, 2024.

[3] Anaïs Halin, Sébastien Piérard, Anthony Cioppa, and Marc Van Droogenbroeck, "A hitchhiker's guide to understanding performances of two-class classifiers," *arXiv*, vol. abs/2412.04377, 2024.

[4] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *IEEE Int. Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Providence, RI, USA, Jun. 2012, pp. 1–8.

[5] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, Columbus, OH, USA, Jun. 2014, pp. 393–400.

[6] Anaïs Halin, Sébastien Piérard, Renaud Vandeghen, Benoît Gérin, Maxime Zanella, Martin Colot, Jan Held, Anthony Cioppa, Emmanuel Jean, Gianluca Bontempi, Saïd Mahmoudi, Benoît Macq, and Marc Van Droogenbroeck, "Physically interpretable probabilistic domain characterization," in *Asian Conf. Comput. Vis. Work. (ACCV Work.)*, 2025, vol. 15482 of *Lect. Notes Comput. Sci.*, pp. 17–35.

[7] Kyle Cranmer, Johann Brehmer, and Gilles Louppe, "The frontier of simulation-based inference," *Proc. National Acad. Sci. (PNAS)*, vol. 117, no. 48, pp. 30055–30062, May 2020.

[8] Thierry Bouwmans, Sajid Javed, Maryam Sultana, and Soon Ki Jung, "Deep neural network concepts for background subtraction:a systematic review and comparative evaluation," *Neural Networks*, vol. 117, pp. 8–66, Sept. 2019.

[9] Thierry Bouwmans and Belmar Garcia-Garcia, "Visual surveillance of human activities: Background subtraction challenges and methods," in *From Visual Surveillance to Internet of Things: Technology and Applications*, pp. 1–24. Taylor & Francis Group, Oct. 2019.

[10] Thierry Bouwmans and Belmar Garcia-Garcia, "Visual surveillance of natural environments: Background subtraction challenges and methods," in *From Visual Surveillance to Internet of Things: Technology and Applications*, pp. 1–20. Taylor & Francis, Oct. 2019.

[11] Marc Braham and Marc Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *IEEE Int. Conf. Syst. Signals Image Process. (IWSSIP)*, Bratislava, Slovakia, May 2016, pp. 1–4.

[12] Olivier Barnich and Marc Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[13] Anthony Cioppa, Marc Van Droogenbroeck, and Marc Braham, "Real-time semantic background subtraction," in *IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab. Emir., Oct. 2020, pp. 3214–3218.

[14] Jérôme Leens, Sébastien Piérard, Olivier Barnich, Marc Van Droogenbroeck, and Jean-Marc Wagner, "Combining color, depth, and motion for video segmentation," in *Computer Vision Systems*. 2009, vol. 5815 of *Lect. Notes Comput. Sci.*, pp. 104–113, Springer.

[15] Pierre-Marc Jodoin, Sébastien Piérard, Yi Wang, and Marc Van Droogenbroeck, "Overview and benchmarking of motion detection methods," in *Background Modeling and Foreground Detection for Video Surveillance*, chapter 24. Chapman and Hall/CRC, Jul. 2014.

[16] Sébastien Piérard, Marc Braham, and Marc Van Droogenbroeck, "An exploration of the performances achievable by combining unsupervised background subtraction algorithms," *arXiv*, vol. abs/2202.12563, 2022.

[17] Marc Braham, Sébastien Piérard, and Marc Van Droogenbroeck, "Semantic background subtraction," in *IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sept. 2017, pp. 4552–4556.

[18] Sébastien Piérard and Marc Van Droogenbroeck, "A perfect estimation of a background image does not lead to a perfect background subtraction: analysis of the upper bound on the performance," in *Int. Conf. Image Anal. Process. (ICIAP), Work. Scene Backgr. Model. Initial. (SBMI)*. 2015, vol. 9281 of *Lect. Notes Comput. Sci.*, pp. 527–534, Springer.

[19] Sébastien Piérard and Marc Van Droogenbroeck, "Summarizing the performances of a background subtraction algorithm measured on several videos," in *IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3234–3238.

[20] David Cossock and Tong Zhang, "Subset ranking using regression," in *Learn. Theory*, 2006, vol. 4005 of *Lect. Notes Comput. Sci.*, pp. 605–619.

[21] Ping Li, Qiang Wu, and Christopher Burges, "McRank: Learning to rank using multiple classification and gradient boosting," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, Can., Dec. 2007, vol. 20, pp. 1–8, Curran Assoc. Inc.

[22] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender, "Learning to rank using gradient descent," in *Int. Conf. Mach. Learn. (ICML)*, Bonn, Germany, 2005, pp. 89–96, ACM Press.

[23] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, "Learning to rank: from pairwise approach to listwise approach," in *Int. Conf. Mach. Learn. (ICML)*, Corvalis, OR, USA, Jun. 2007, p. 129–136, ACM.

[24] T. Liu, "Learning to rank for ir," *Foundation and Trends in IR*, vol. 3, no. 3, pp. 225–331, 2009.

[25] Niek Tax, Sander Bockting, and Djoerd Hiemstra, "A cross-benchmark comparison of 87 learning to rank methods," *Inf. Process. & Manag.*, vol. 51, no. 6, pp. 757–772, Nov. 2015.

[26] MMSegmentation Contributors, "MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Scene parsing through ADE20K dataset," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5122–5130.

[28] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "COCO-stuff: Thing and stuff classes in context," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 1209–1218.

[29] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 1280–1289.

APPENDIX

A. Useful resources

Useful resources can be found at:

https://github.com/pierard/performance

B. Enlarged version of Fig. 3

Fig. 9 is an enlarged version of Fig. 3 on which it is easier to observe the 40 layers of the mille-feuilles.

C. BGS methods considered in Sec. IV

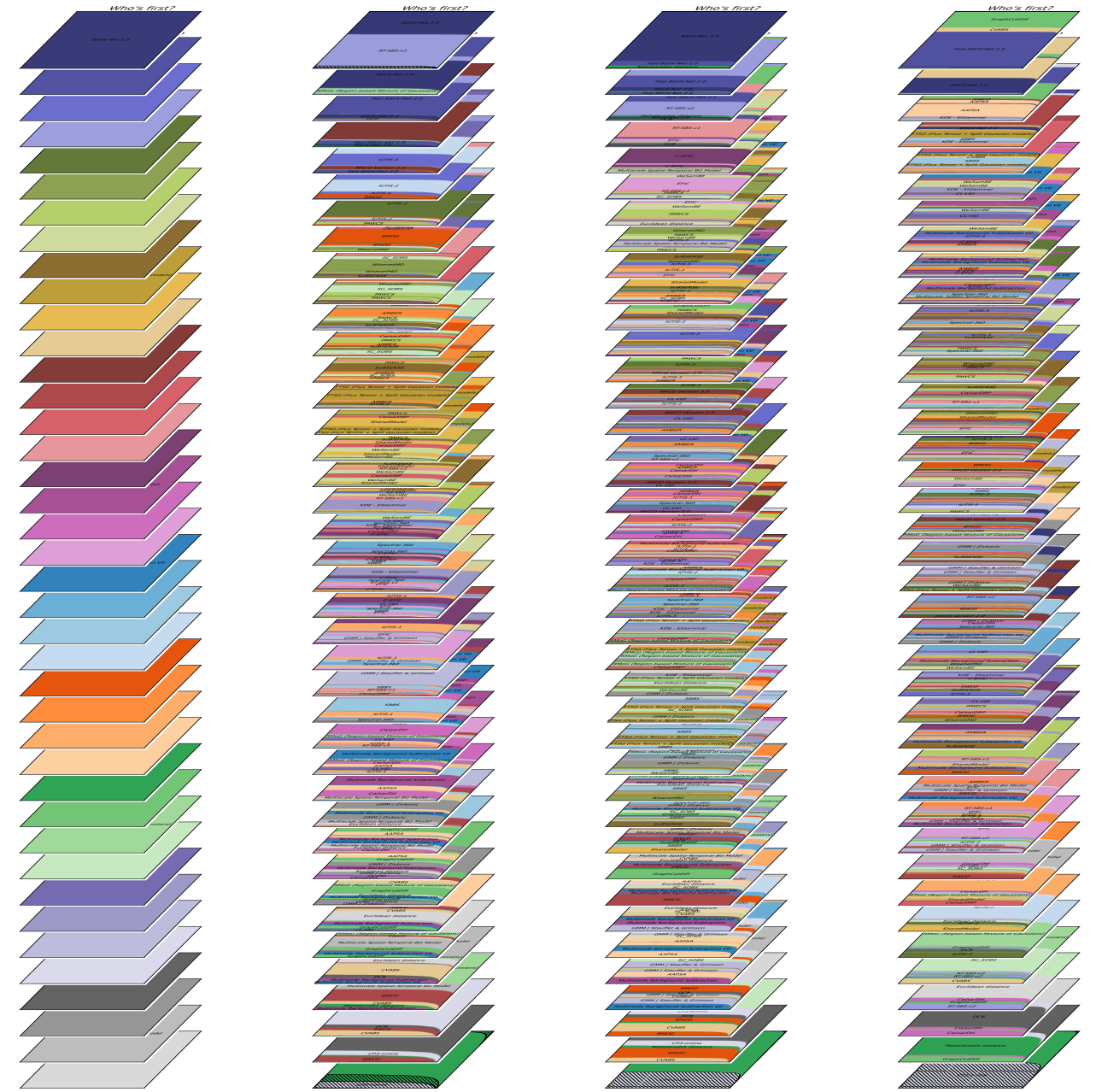
We considered 40 unsupervised BGS methods. More precisely, we selected those for which the output masks were available on changedetection.net, for all videos of *CDnet 2014*, on March 11th 2025, and excluded those tagged as "supervised" on the platform.

1) BSUV-Net 2.0
2) Fast BSUV-Net 2.0
3) IUTIS-5
4) RT-SBS-v2
5) IUTIS-3
6) WisenetMD
7) PAWCS
8) WeSamBE
9) SuBSENSE
10) FTSG (Flux Tensor with Split Gaussian mdoels)
11) SharedModel
12) CVABS
13) M4CD Version 2.0
14) SWCD
15) CwisarDRP
16) RT-SBS-v1
17) C-EFIC
18) Multimode Background Subtraction
19) CwisarDH
20) EFIC
21) Multimode Background Subtraction Version 0 (MBS V0)
22) Spectral-360
23) Sample based background subtractor (SBBS)
24) IUTIS-2
25) BMOG
26) AMBER
27) IUTIS-1
28) AAPSA
29) Mahalanobis distance
30) GraphCutDiff
31) RMoG (Region-based Mixture of Gaussians)
32) SC_SOBS
33) CL-VID
34) KDE - ElGammal
35) GMM | Stauffer & Grimson
36) CP3-online
37) DCB
38) GMM | Zivkovic
39) Multiscale Spatio-Temporal BG Model
40) Euclidean distance

D. Detailed results for the first experiment

For the 11 strategies involved in our 1st experiment, we provide the correlation Tiles for each domain (the 53 videos of *CDnet 2014*), as well as the average (mean correlation Tile) and worst-case (minimum correlation Tile) over all domains. These are the intermediate results that are summarized in Fig. 6.

- Fig. 10: detailed results for CDnet
- Fig. 11: detailed results for CDnet*
- Fig. 12: detailed results for mean-P
- Fig. 13: detailed results for mean-P*
- Fig. 14: detailed results for sem-P†
- Fig. 15: detailed results for sem-P$^{\dagger *}$
- Fig. 16: detailed results for $\mathrm{sem-d}^\dagger$
- Fig. 17: detailed results for $\mathrm{sem-d}^{\dagger *}$
- Fig. 18: detailed results for avg†
- Fig. 19: detailed results for avg$^{\dagger *}$
- Fig. 20: detailed results for all†



(a) Strategy CDnet (b) Strategy $\mathrm{sem} - \mathrm{d}^{\dagger}$ (c) Strategy $\mathrm{sem} - \mathrm{d}^{\dagger *}$ (d) Real rankings

(e) Strategy CDnet (f) Strategy $\mathrm{sem} - \mathrm{d}^{\dagger}$ (g) Strategy $\mathrm{sem} - \mathrm{d}^{\dagger *}$ (h) Choice *w.r.t.* I.

Fig. 9: Enlarged version of Fig. 3.

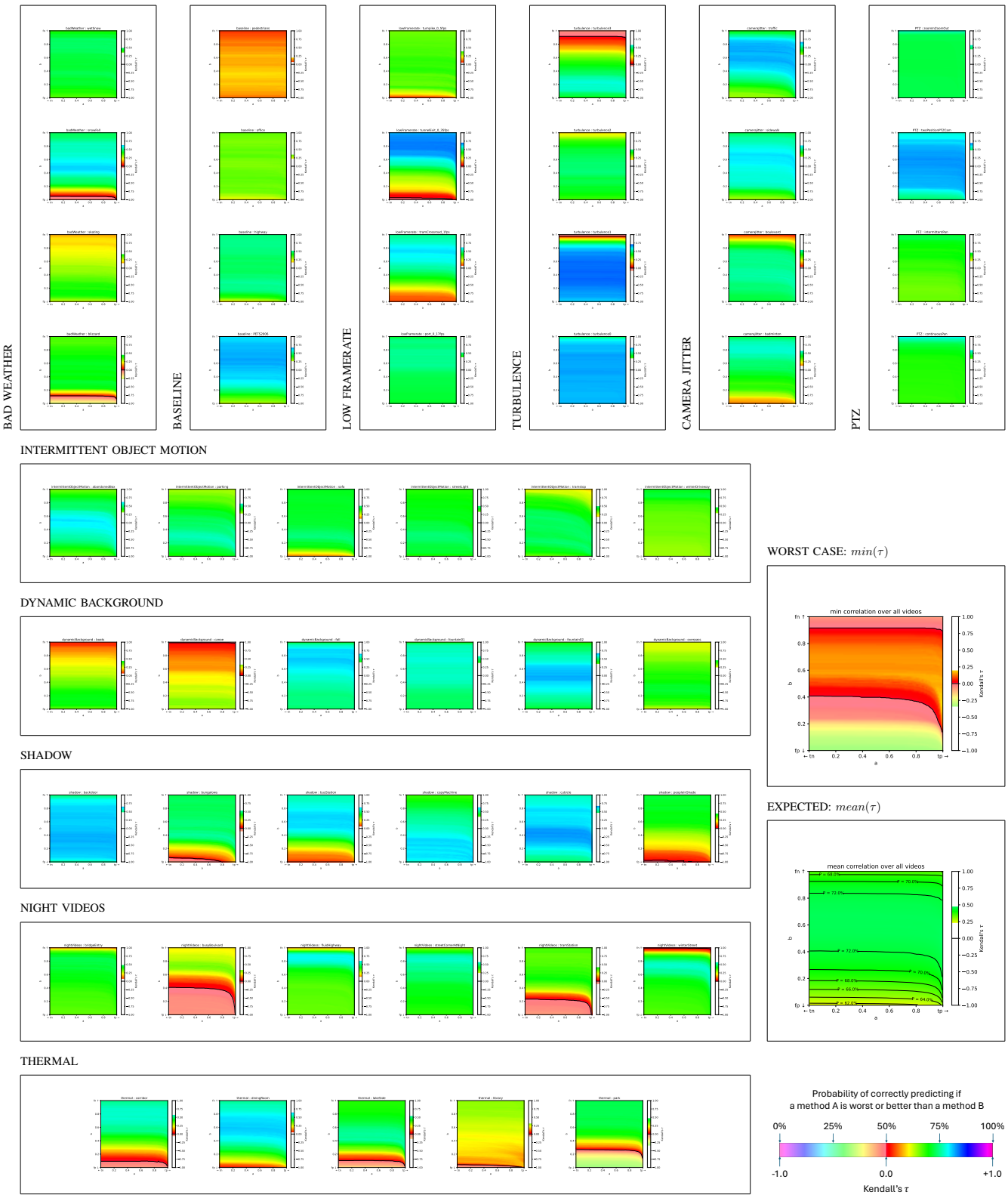


Fig. 10: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "CDnet", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

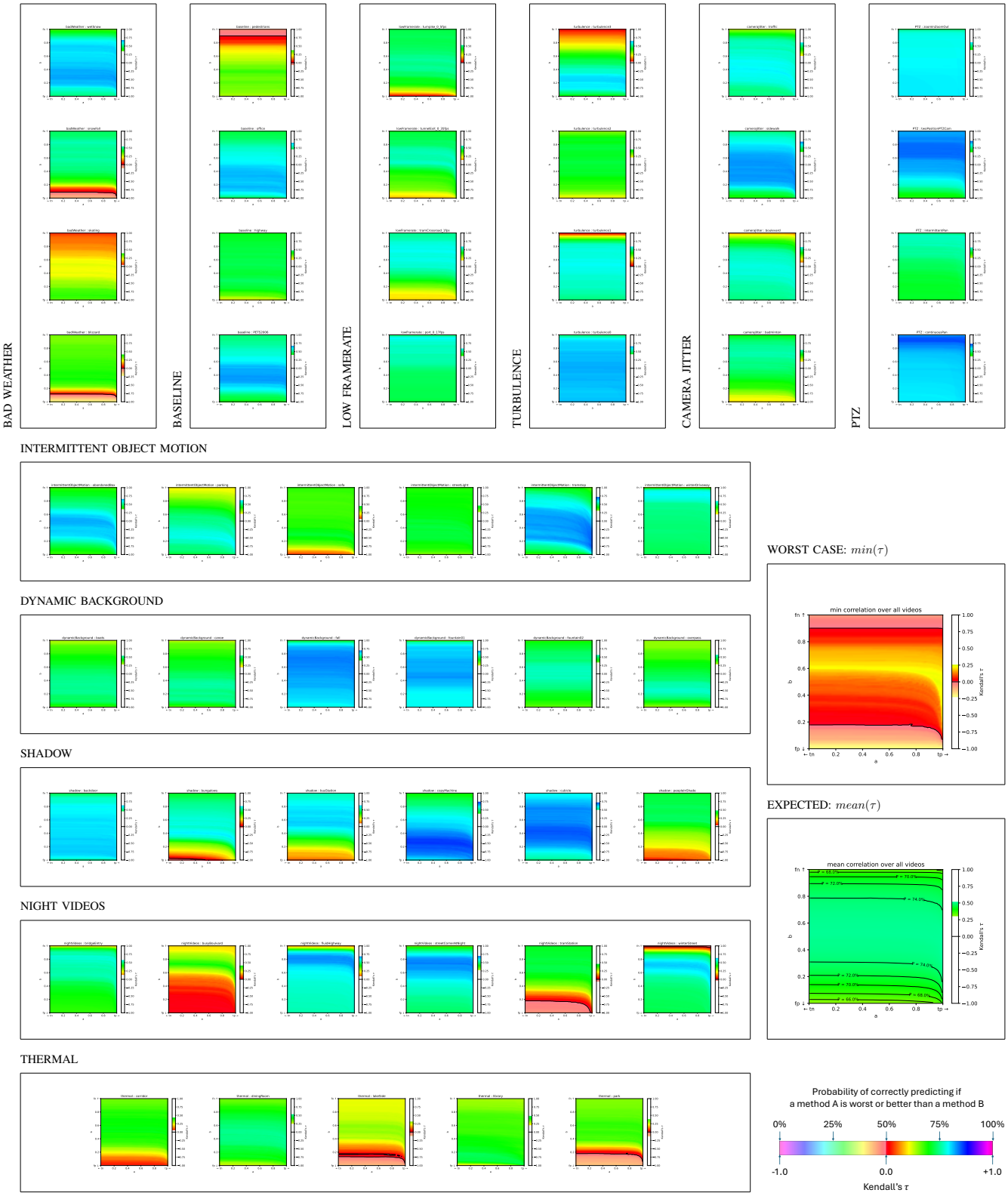


Fig. 11: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "CDnet*", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

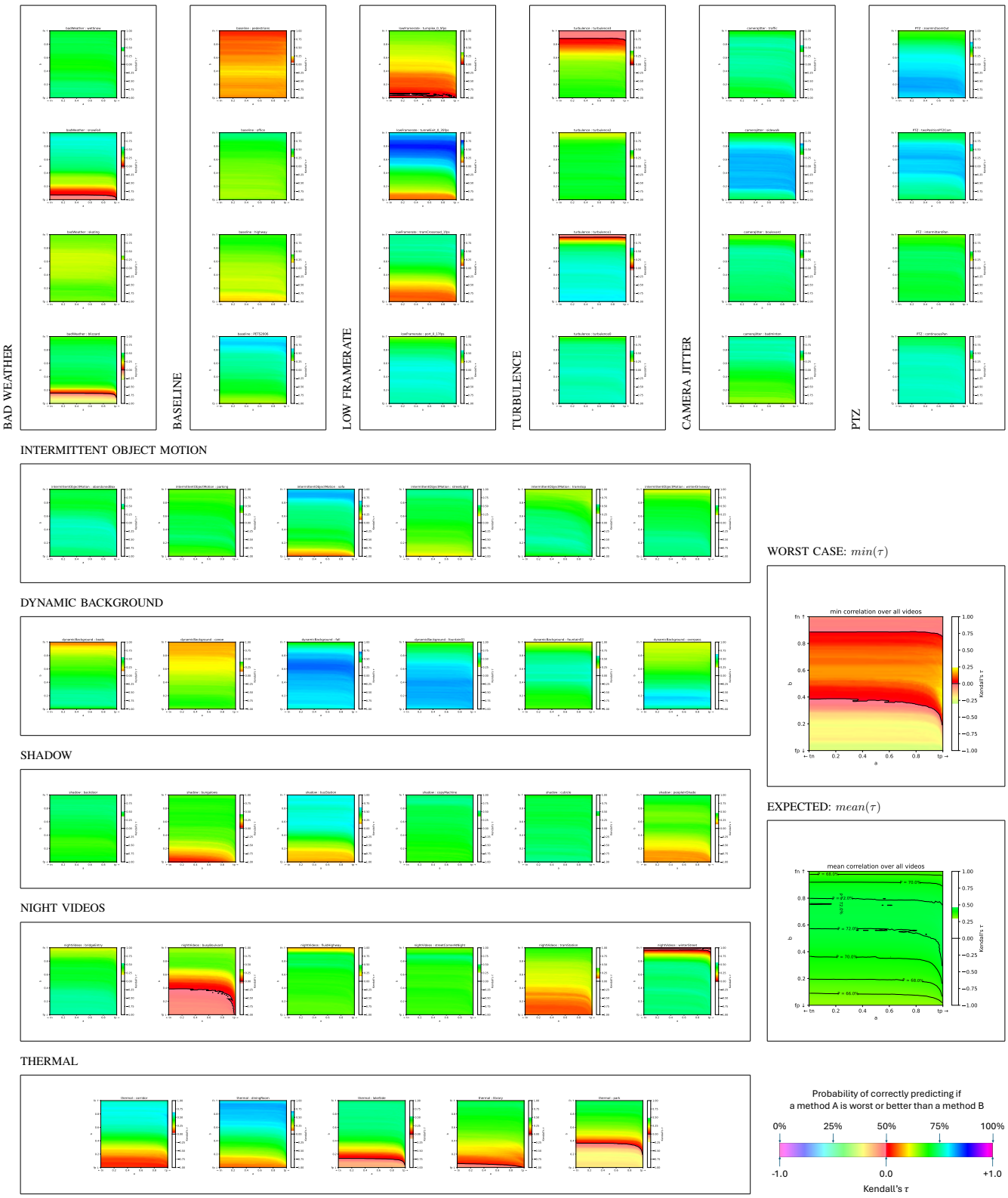


Fig. 12: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "mean-P", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

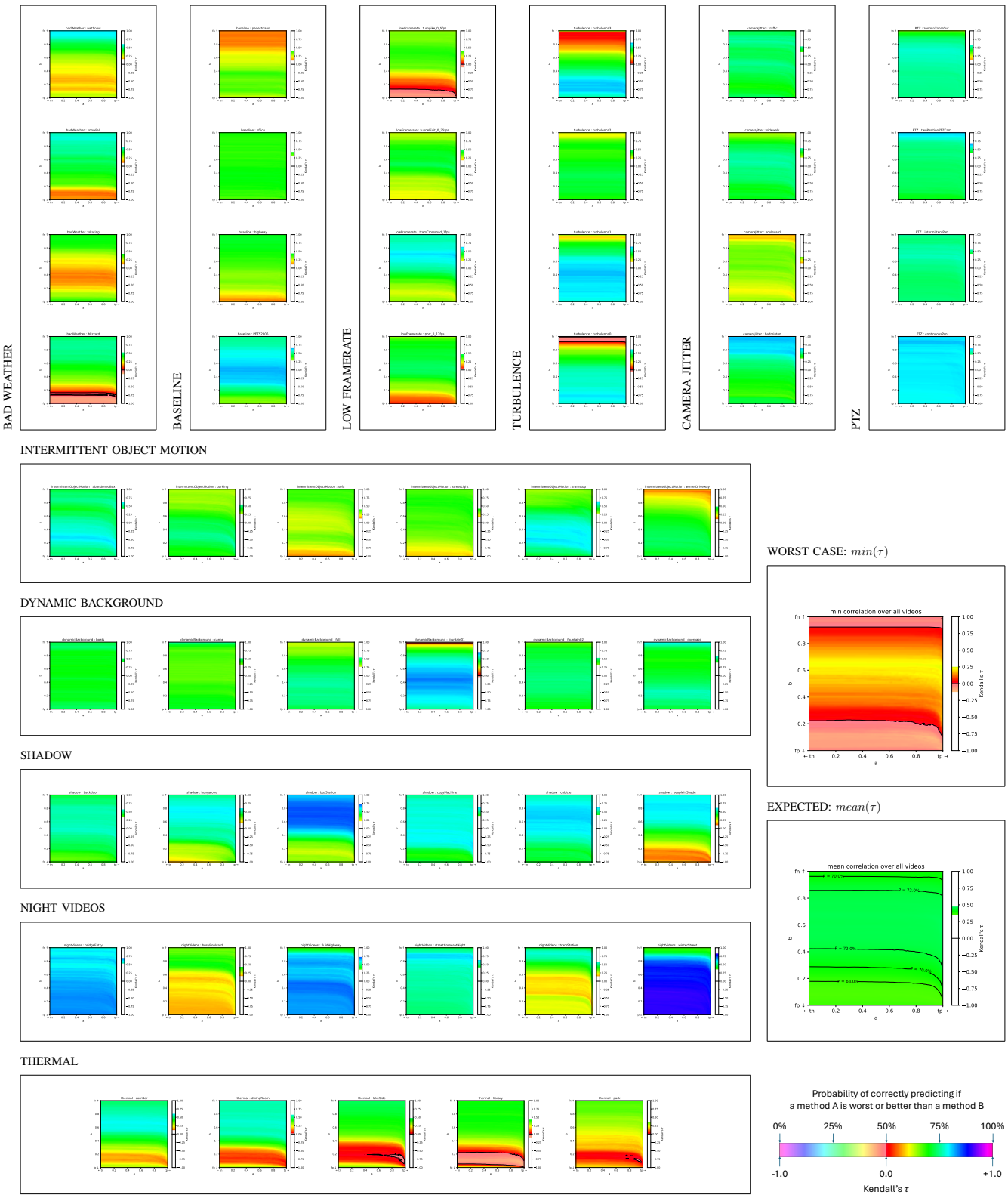


Fig. 13: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "mean-P*", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

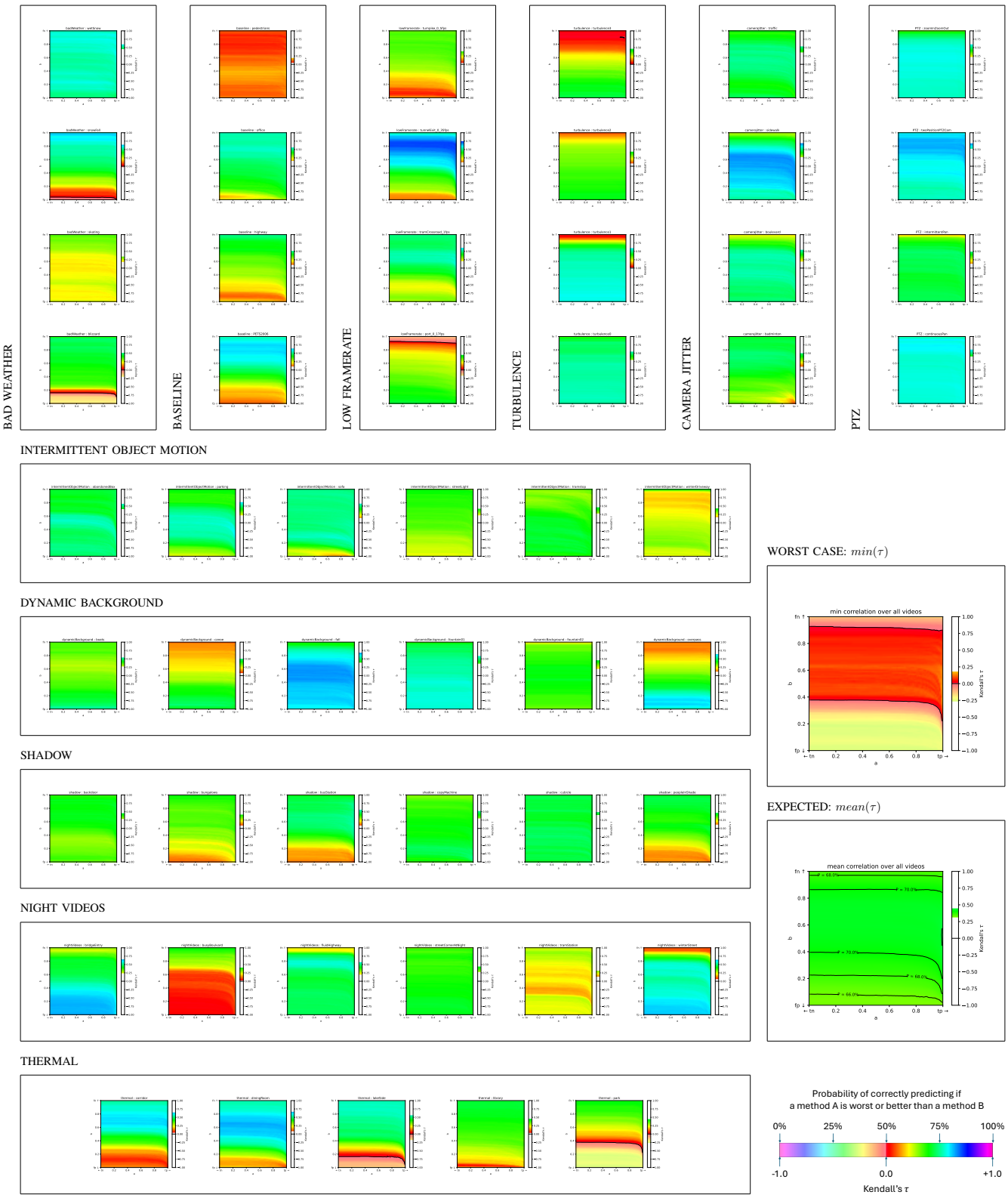


Fig. 14: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "sem-P†", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

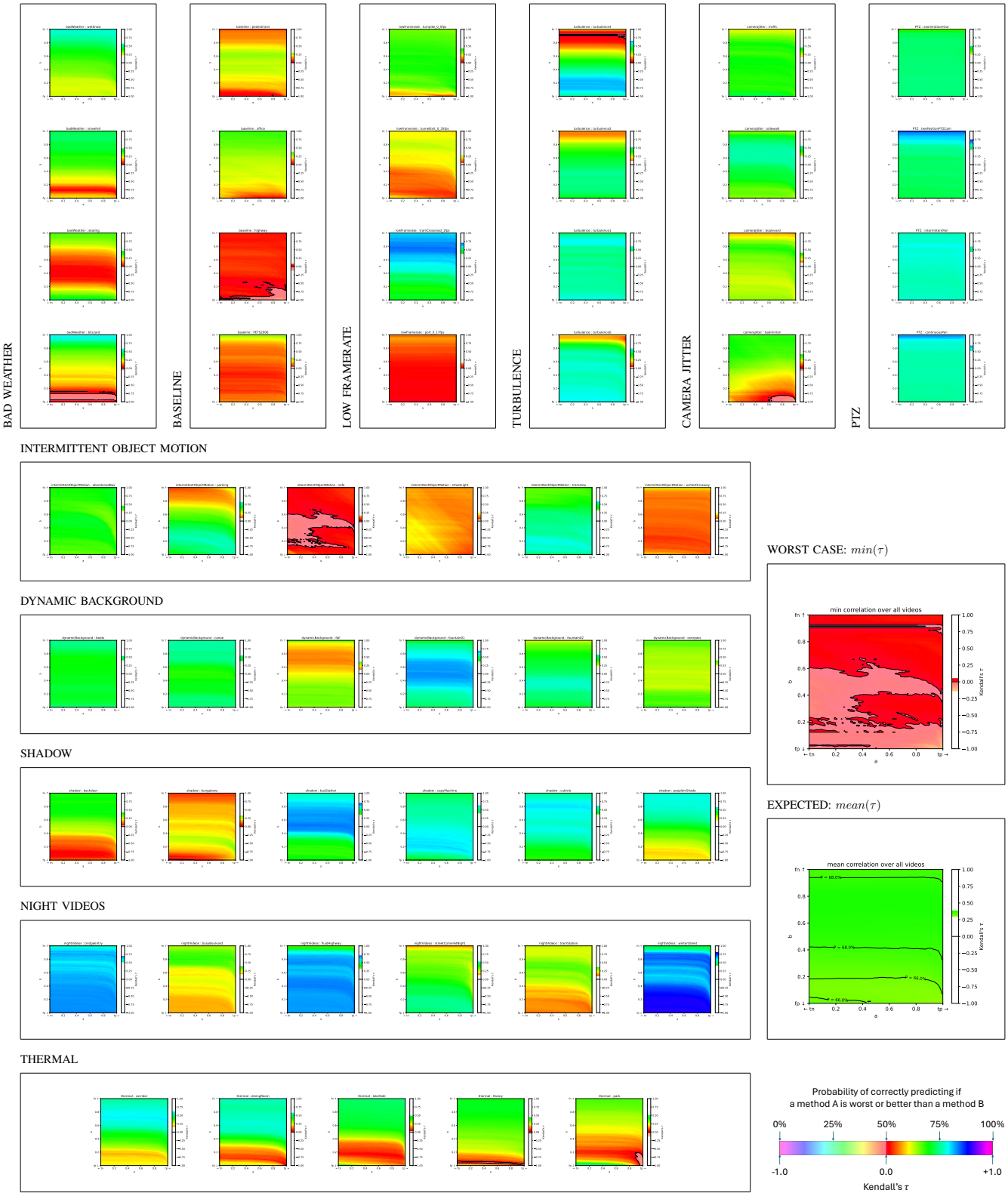


Fig. 15: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "sem-P$^{\dagger *}$", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

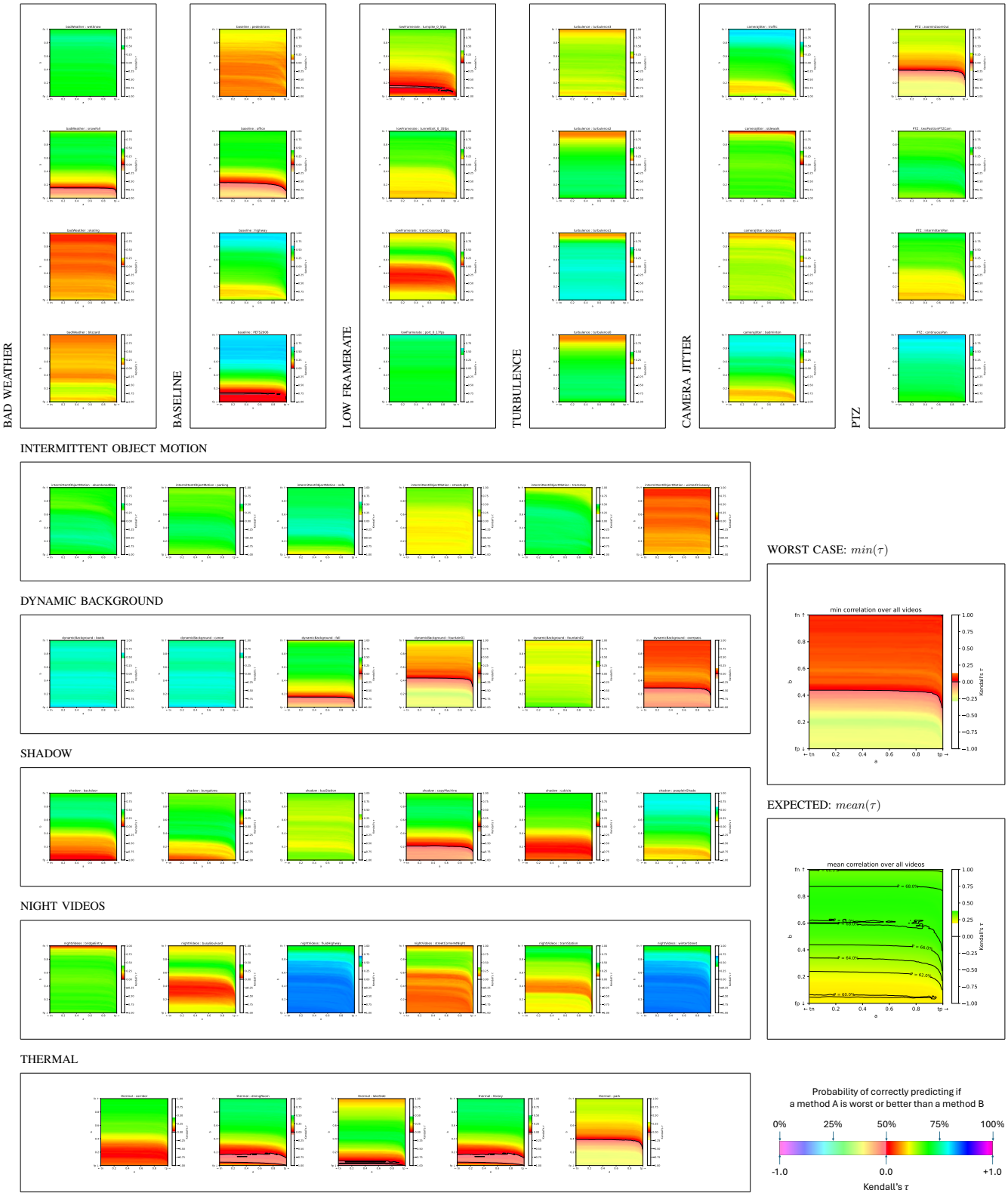


Fig. 16: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "$\mathrm{sem} - \mathrm{d}^{\dagger}$", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

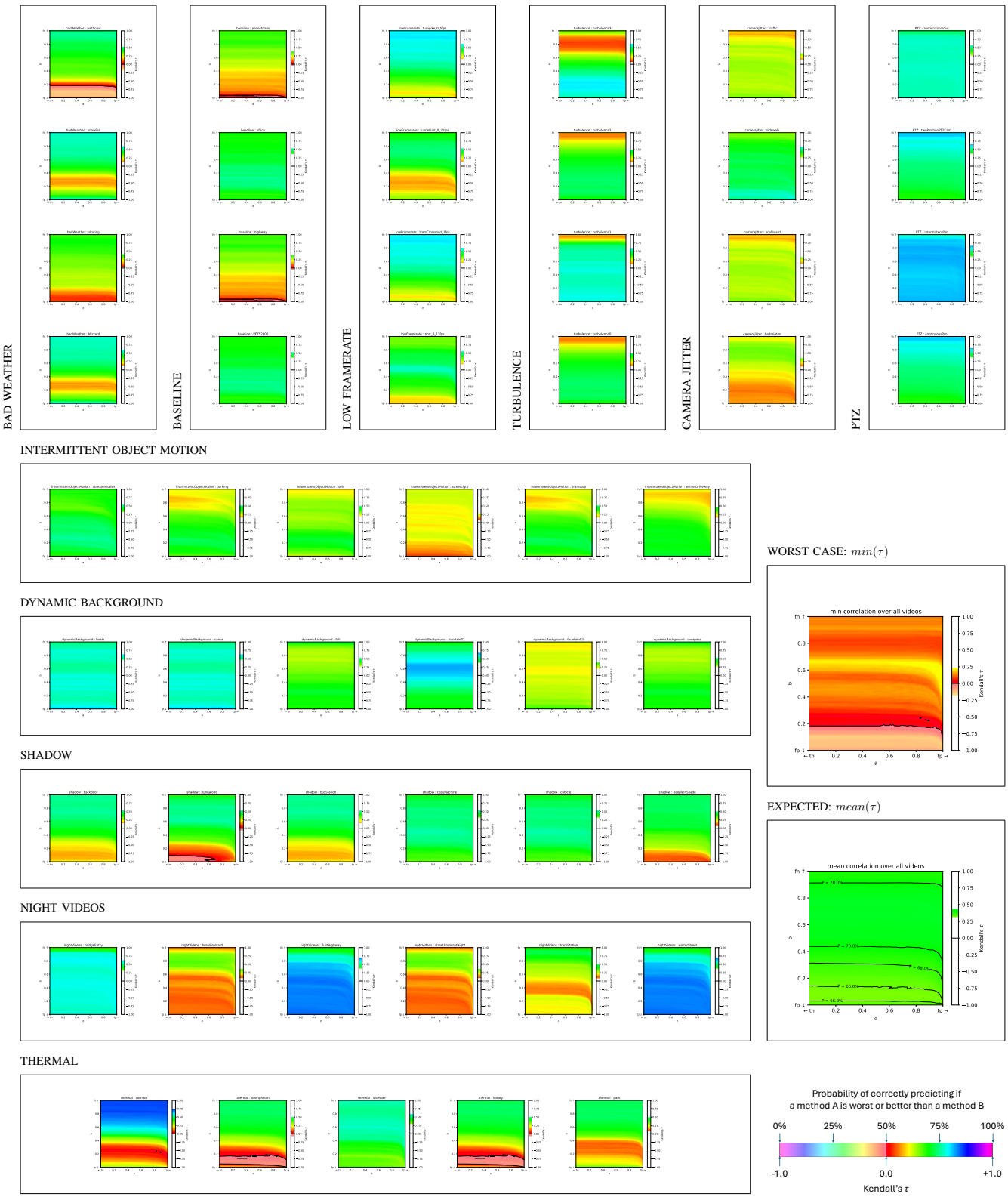


Fig. 17: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "$\mathrm{sem} - \mathrm{d}^{\dagger *}$", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

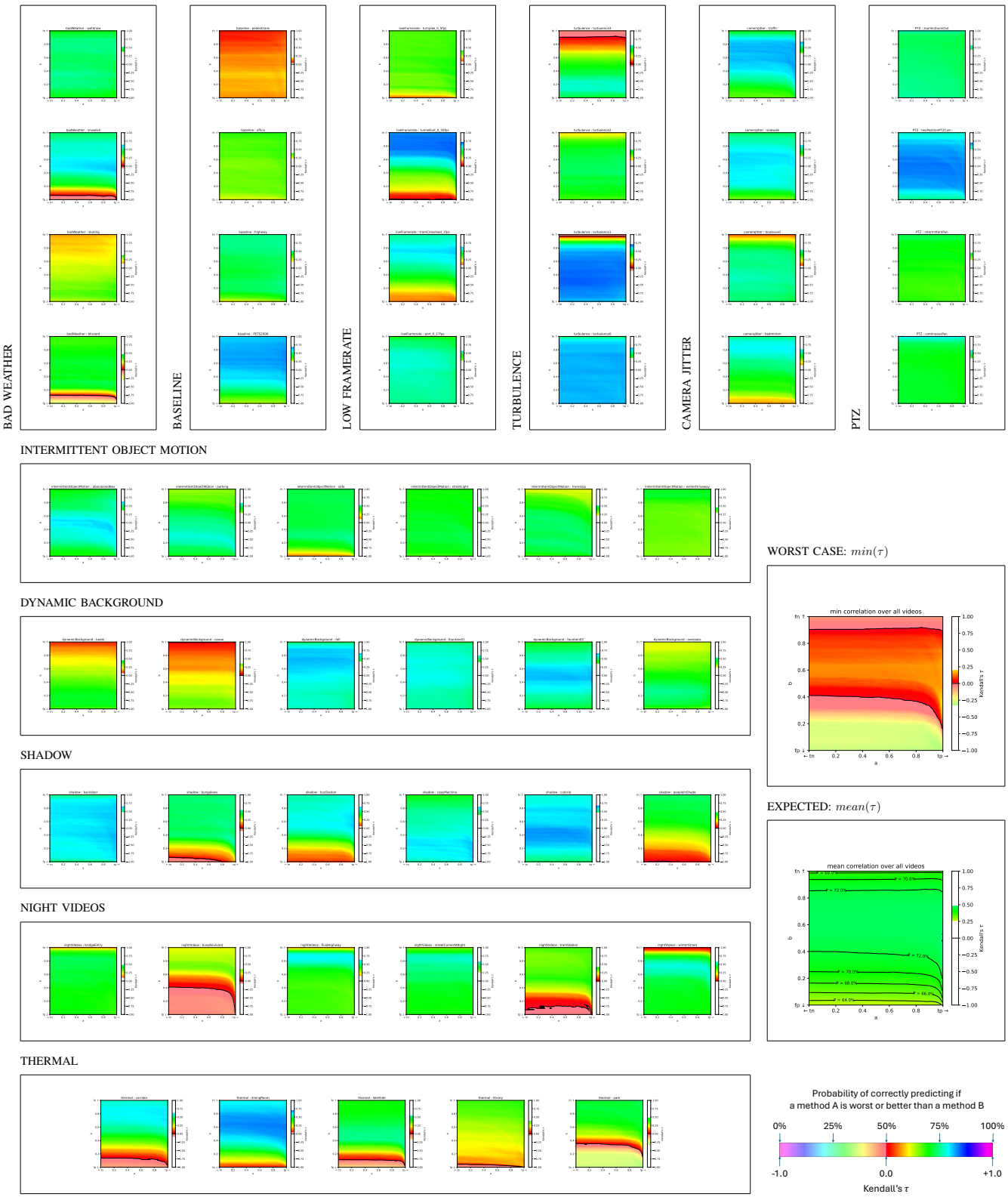


Fig. 18: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "avg†", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

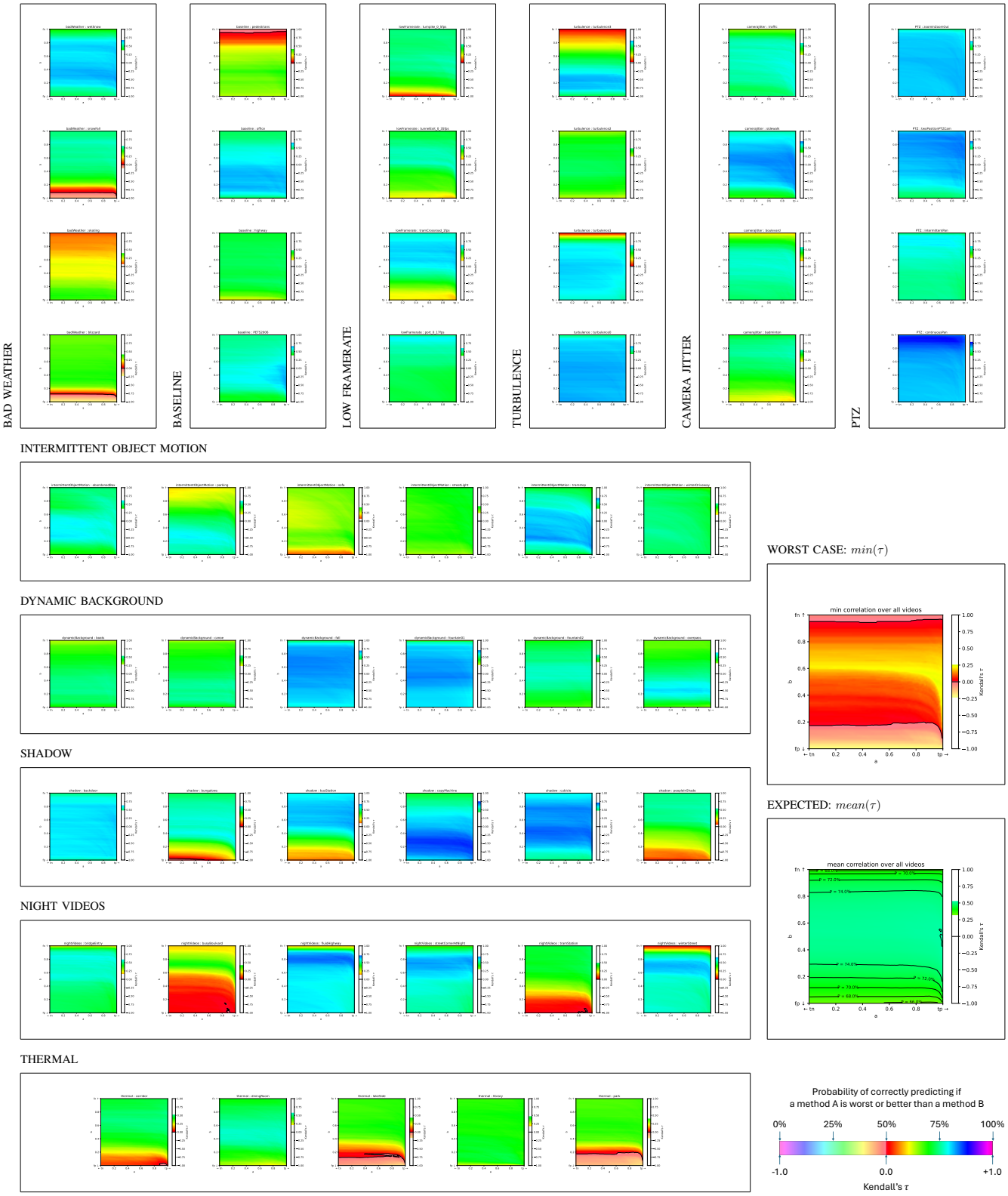


Fig. 19: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "avg$^{\dagger *}$", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.

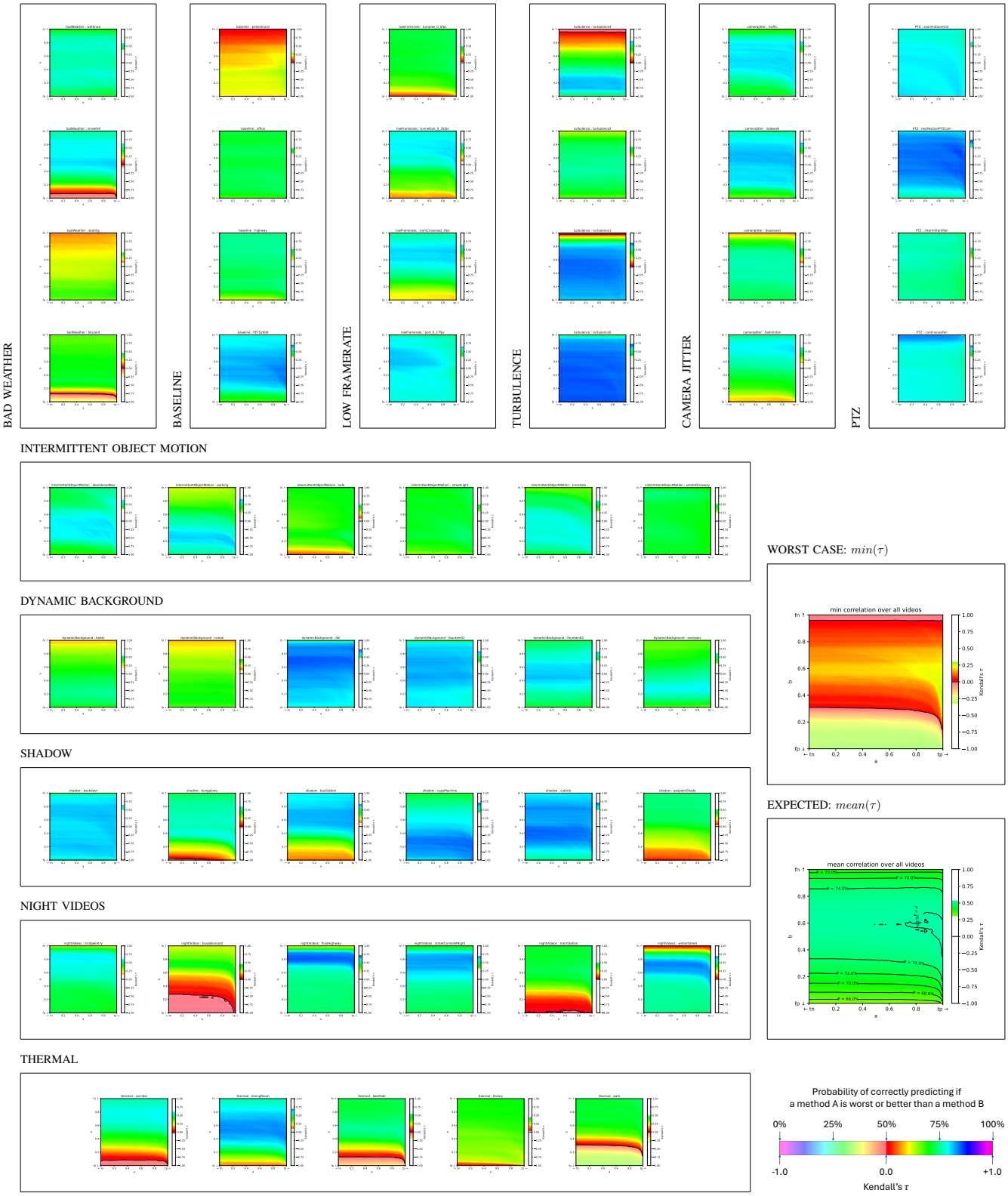


Fig. 20: Tiles showing the correlations (Kendall's τ), *w.r.t.* the application-specific preferences I, between the true ranking and the one predicted with the strategy "all†", for 53 videos, as well as in the behavior in the worst case and in average. These rankings involve 40 BGS methods.