A Hitchhiker's Guide to Understanding Performances of Two-Class Classifiers

Anaïs Halin*, Sébastien Piérard*, Anthony Cioppa, and Marc Van Droogenbroeck
Montefiore Institute, University of Liège, Liège, Belgium
{Anais.Halin,S.Pierard,Anthony.Cioppa,M.VanDroogenbroeck}@uliege.be

Abstract

Properly understanding the performances of classifiers is essential in various scenarios. However, the literature often relies only on one or two standard scores to compare classifiers, which fails to capture the nuances of application-specific requirements, potentially leading to suboptimal classifier selection. Recently, a paper on the foundations of the theory of performance-based ranking introduced a tool, called the Tile*, that organizes an infinity of ranking scores into a 2D map. Thanks to the Tile, it is now possible to evaluate and compare classifiers efficiently, displaying all possible application-specific preferences instead of having to rely on a pair of scores. In this paper, we provide a first hitchhiker's guide for understanding the performances of two-class classifiers by presenting four scenarios, each showcasing a different user profile: a theoretical analyst, a method designer, a benchmarker, and an application developer. Particularly, we show that we can provide different interpretative flavors that are adapted to the user's needs by mapping different values on the Tile. As an illustration, we leverage the newly introduced Tile tool and the different flavors to rank and analyze the performances of* 74 *state-of-the-art semantic segmentation models in two-class classification through the eyes of the four user profiles. Through these user profiles, we demonstrate that the Tile effectively captures the behavior of classifiers in a single visualization, while accommodating an infinite number of ranking scores.* [1]

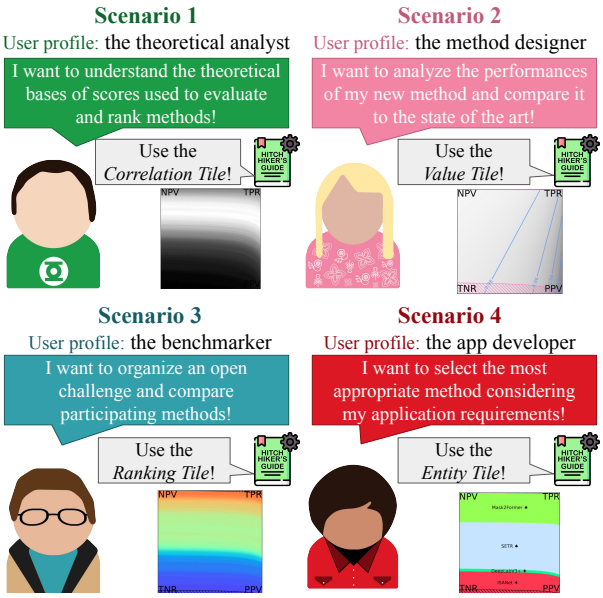


Figure 1. **Our hitchhiker's guide.** This hitchhiker's guide to understanding performances of two-class classifiers addresses four scenarios, answering specific requests from four user profiles: (1) *the theoretical analyst*, who is interested in understanding the theoretical relationship between different scores typically used for evaluating or ranking methods, (2) *the method designer*, who would like to analyze the performances of his/her new method and compare it to others, (3) *the benchmarker*, who organizes challenges for the scientific community and would like to know how to rank participating methods, and finally (4) *the application developer*, who wants to select the most appropriate method for his/her application. This guide provides specific tools and explains how to interpret them for each of those four scenarios.

1. Introduction

As humans, performance and ranking are widespread in all aspects of our lives. For instance, in school, teachers evaluate tests and homework using a score which reflects the performance of students. In some disciplines such as calculus, evaluation is straightforward as there are only two possible cases: either the answer is correct or wrong. The

*Equal contributions.

[1] This paper is the third of a trilogy. In a nutshell, paper A [44] presents an axiomatic framework and an infinite family of scores for ranking classifiers. In paper B [45], we particularize this framework to binary classification and present the *Tile* that organizes these scores (among which PPV, TPR, TNR, F_1, and A) in a single plot. Finally, this paper (paper C [26]) provides a guide to using the Tile according to four practical scenarios. For that, we present different Tile flavors that are applied to a real case.

score can then be calculated as the ratio of correct answers to the total number of questions, which is straightforward. Likewise, in most team sports, team A beats team B if they score more points. Even for reviewing papers, area chairs use scores provided by the reviewers to assess if a paper should be accepted to or rejected from the conference [38].

However, not all evaluations are well-defined. For instance, when grocery shopping, consumers may choose product A over product B looking at different characteristics such as the price, the amount of sugar, or the packaging. In this case, the choice is based on several, sometimes contradictory scores. The question is, therefore, which score should the choice be based on? Similar questions arise in the field of machine learning: How can we determine if a newly designed classifier outperforms existing ones? Which score(s) should we use to analyze its performance? How do we decide which score to consider for ranking the classifiers? Answers to these questions are not trivial, while potentially having a big impact on the development of a whole field of research.

In this paper, we propose a step-by-step hitchhiker's guide to help compare, analyze, and rank two-class classifiers. Throughout this guide, we study four scenarios, answering specific requests that four common user profiles may have, and provide visual tools adapted to their needs, as illustrated in Fig. 1. To do so, we rely on the infinite and parametric family of scores satisfying an axiomatic definition of the performance-based rankings introduced in paper A [44] and the tool called the *Tile* introduced in paper B [45]. More precisely, we leverage the newly introduced Tile tool to present different flavors for visualizing various elements useful for understanding the performances of two-class classifiers under different angles and considering an infinite number of scores. As such, we propose several ways to construct, use, and interpret the Tile in the four scenarios, offering practical guidance to different user profiles with varied objectives, such as theoretically analyzing scores, developing new methods, benchmarking challenges, or designing applications. Finally, we show that the Tile is a versatile tool that can serve a wide range of people in the field of computer vision through an illustration that analyzes and ranks 74 state-of-the-art semantic segmentation models trained on 4 datasets.

Contributions. We summarize our contributions as follows. **(i)** We provide the first hitchhiker's guide to understanding the performances of two-class classifiers, anchored in rigorous theoretical foundations. **(ii)** Throughout the guide, we answer the specific needs of four common user profiles via four scenarios by detailing which tool they should use, how they are constructed, and how they should interpret the results. **(iii)** We illustrate our guide for the computer vision community with an analysis and ranking of 74 state-of-the-art semantic segmentation models.

2. Related Work

In general terms, Japkowicz and Shah [33] decompose performance evaluation and analysis into four parts: (1) the analysis of performance measures, (2) the estimation of errors, (3) the study of statistical testing, and (4) understanding of the experimental power of datasets. In machine learning, the importance of these parts is related to the application. For example, when datasets are too small, the amount of uncertainty in the measured performances should be scrutinized with tools that examine their statistical significance [18]. However, statistical significance is less of an issue in the field of computer vision, as it commonly disposes of enough samples to evaluate a classifier. For the analysis of performance measures, which is the focus of our guide, two strategies have emerged.

The first strategy consists in providing a series of performance scores that highlight the benefits and trade-offs in designing a classifier. Tharwat [52] analyzed a series of scores for classification tasks that can be used alone or in combination. Hereafter, we discuss some of these scores. For example, the weighted harmonic mean between the recall TPR and the precision PPV, denoted by F_β, or simply by F_1 when weights are equal, is advocated by many authors. A score similar to F_1 is Jaccard's coefficient. Yet, it is known that Jaccard's coefficient and the F_1 score lead to the same ranking of classifiers, although Jaccard is probabilistic, unlike F_1 [23, 47]. The unweighted arithmetic mean of TPR and TNR is the balanced accuracy. It is considered as the most suitable score for evaluating imbalanced problems, thanks to its independence regarding the priors [3, 7, 25, 36]. However, the problem with the first strategy is that even though we can use some scores for analyzing the performances of classifiers and ranking them, it is not clear which ones to use.

The second strategy makes use of so-called evaluation spaces, such as the *Receiver Operating Characteristic* (ROC) [41] or *Precision-Recall* (PR) spaces, that combine two scores. The ROC space is a plot of sensitivity vs. one minus specificity as one varies a cutoff on a continuous predictor used to decide [27]. In [21, 22], Fawcett explained how to use ROC graphs and avoid interpretation pitfalls. For instance, the area under the ROC curve (AUC-ROC) is commonly used as a performance indicator, as it has statistical significance [2]. Although several authors raised concerns about ROC-derived scores in decision-making [6, 10], recent works such as the one on contingency spaces [1] still build upon ROC graphs. More recently, the PR space has become the de facto replacement for the ROC space in the presence of imbalance [32], ignoring the fact that there is a bijection between these two spaces [17]. Moreover, the PR space suffers from the presence of an unachievable region [4], which is often overlooked in practice. The primary drawback of ROC- or PR-based curves analyzes is that, ac-

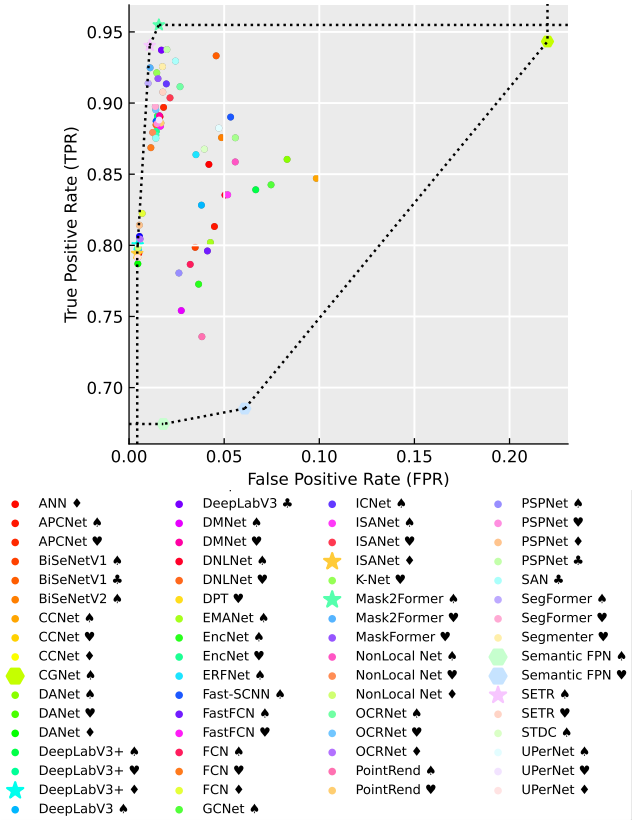


Figure 2. **How can we rank all these classification performances using the ROC space?** The classical ROC space does not provide an answer at a glance. The 74 performances shown here (for a positive prior equal to 0.124) serve as the showcase for our illustration in this guide. The dash lines correspond to the supremum and infimum of all achievable performances using a combination of the classifiers.

cording to Menon *et al*. [40], the AUC-ROC and AUC-PR apply not to a classifier but to a scoring function, and that a scoring function yields a family of classifiers obtained for different thresholds. In other words, these curves consider a parametric family of classifiers rather than a specific classifier, and they are therefore inadequate to select classifiers when it comes to comparing unique instances from different families. As an example of these drawbacks, we show the performance of 74 two-class classifiers in the ROC space in Fig. 2. As can be seen, it is extremely challenging to uniquely determine which classifier is the best or the worst. Moreover, it is even harder to rank all 74 classifiers based only on this ROC space. For instance, relying on a single of the two presented scores such as the True Positive Rate (TPR) may result in overfitting on that score, disregarding other important nuances of the performance.

While these strategies provide a good insight from multiple perspectives, they only offer a scattered interpretation due to the inherent partial redundancy between scores, and ultimately only offer an incomplete look on the performances and ranking. In a recent attempt to formalize the notion of ranking, Nguyen *et al*. [42] proposed to impose three properties for ranking: (1) reliability, (2) meaningfulness (evaluated by humans), and (3) mathematical consistency. In paper A [44], we present an alternative formal axiomatic definition of performance-based rankings, grounded in order theory and anchored within a probabilistic framework. The axioms ensure the stability of rankings, *i.e*., that if multiple entities, in our case two-class classifiers, are ranked, adding or removing an entity does not affect the relative order of the previously present entities. Additionally, it introduces ranking scores that satisfy these axioms, parameterized by a random variable I, called *importance*, which allows for consideration of application-specific preferences. Paper B [45] proposes a spatial organization of the ranking scores in a 2D square map, called the Tile, for the particular case of two-class classification. It also studies properties of the Tile from a theoretical perspective. In this work, we show how both the axiomatic framework and the Tile can be used in practice and propose a hitchhiker's guide to use the Tile in various ways through four scenarios and user profiles.

3. Hitchhiker's Guide

In this section, we present our hitchhiker's guide to understanding performances of two-class classifiers. We begin by providing the recommended basic theoretical knowledge for using the guide, including a description of the terminology and notations, the general Tile tool, and the illustrative context used throughout. Next, we detail four scenarios, each aligned with the needs of a specific user profile. For each scenario, we provide the context, highlight the relevant flavors of the Tile, and explain how to interpret the results.

3.1. Recommended basic theoretical knowledge

Terminology and notations. To be consistent with paper A [44] and paper B [45], we use the same terminology and notations. Hence, an *entity* ϵ is a two-class classifier and the set of all entities of interest is denoted by $\mathbb{E}$. *Performances*, denoted by P, are probability measures, and the performance P_ϵ of an entity ϵ is the evaluation of this entity. *Scores* are functions associating a real value to performances, that is, $X : \mathrm{dom}(X) \to \mathbb{R} : P \mapsto X(P)$, with the domain of the score, $\mathrm{dom}(X)$, included in the set of all probability measures, $\mathbb{P}_{(\Omega,\Sigma)}$, on the measurable space (Ω, Σ), where Ω is the sample space (*i.e*., the set of outcomes) and Σ is the event space (*i.e*., a σ-algebra on Ω).

Construction and interpretation of the Tile. As explained in paper B [45], the Tile for two-class classification is obtained by organizing ranking scores on a 2D square map, where the two axes, a and b, respectively represents the im-

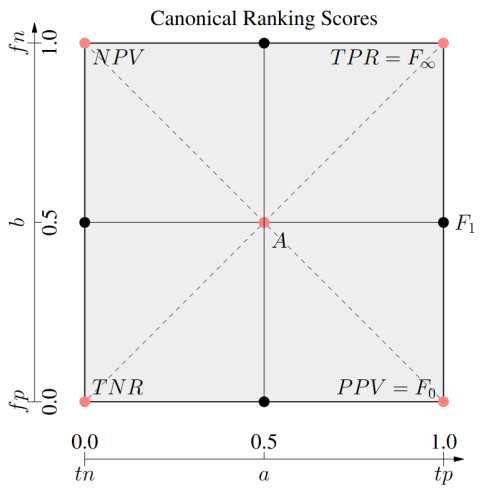


Figure 3. **Tile with canonical ranking scores for two-class classification.** Each point of the Tile corresponds to a ranking score which can be computed using Eq. (3), with the two axes, a and b, defined in Eqs. (1) and (2), respectively. The Tile therefore contains an infinity of ranking scores, including the popular Accuracy (A), Negative Predictive Value (NPV), True Positive Rate (TPR), Positive Predictive Value (PPV), True Negative Rate (TNR), and F_1 scores.

portance I given to true positive (tp) compared to true negative (tn), and false negative (fn) compared to false positive (fp):

$$a = I(tp) = 1 - I(tn)\,, \tag{1}$$

$$b = I(fn) = 1 - I(fp)\,. \tag{2}$$

The importance value can be arbitrarily chosen to reflect application-specific preferences. In the particular case of two-class classification, the ranking scores are given by

$$R_I(P) = \frac{I(tn)P(\{tn\})+I(tp)P(\{tp\})}{I(tn)P(\{tn\})+I(fp)P(\{fp\})+I(fn)P(\{fn\})+I(tp)P(\{tp\})}\,, \tag{3}$$

with $P(\{tp\})$ (resp. $P(\{tn\})$, $P(\{fp\})$, $P(\{fn\})$, and $P(\{tn\})$) corresponding to the probability of event $\{tp\}$ (resp. $\{fp\}$, $\{fn\}$, and $\{tn\}$).

The Tile is then defined as follows:

Definition 1. *The Tile for two-class classification is the mapping*

$$[0,1]^2 \to \mathbb{X}_{(\Omega,\Sigma)} : (a,b) \mapsto R_I$$

with $\mathbb{X}_{(\Omega,\Sigma)}$ denoting all possible scores on (Ω,Σ) and $I(tn) = 1-a$, $I(fp) = 1-b$, $I(fn) = b$, $I(tp) = a$,

The layout of the Tile is shown in Fig. 3. By construction, the top-right corner of the Tile gives maximum importance values to both tp and fn (*i.e.*, $I(tp) = I(fn) = 1$). The ranking score in this corner is thus the True Positive Rate (TPR), also known as recall or sensitivity:

$$TPR = \frac{P(\{tp\})}{P(\{tp\})+P(\{fn\})} = P(Y = \hat{Y}|Y = c_+)\,, \tag{4}$$

where Y is the ground truth, $\hat{Y}$ the prediction, and c_+ the positive class. Conversely, the bottom-left corner gives minimum importance values to both tp and fn (*i.e.*, $I(tp) = I(fn) = 0$), leading to the ranking score named the True Negative Rate (TNR), also known as specificity:

$$TNR = \frac{P(\{tn\})}{P(\{tn\})+P(\{fp\})} = P(Y = \hat{Y}|Y = c_-)\,, \tag{5}$$

where c_- is the negative class. Similarly, the two other corners correspond to the Negative Predictive Value (NPV) and the Positive Predictive Value (PPV), also known as precision:

$$NPV = \frac{P(\{tn\})}{P(\{tn\})+P(\{fn\})} = P(Y = \hat{Y}|\hat{Y} = c_-)\,, \tag{6}$$

$$PPV = \frac{P(\{tp\})}{P(\{tp\})+P(\{fp\})} = P(Y = \hat{Y}|\hat{Y} = c_+)\,. \tag{7}$$

The score in the middle of the Tile, *i.e.*, giving the same importance value to tp, tn, fn, and fp (*i.e.*, $I(tp) = I(tn) = I(fn) = I(fp) = 0.5$) is the popular accuracy A, also defined as $P(Y = \hat{Y})$:

$$A = \frac{0.5\,P(\{tp\})+0.5\,P(\{tn\})}{0.5\,P(\{tp\})+0.5\,P(\{fn\})+0.5\,P(\{fp\})+0.5\,P(\{tn\})}\,. \tag{8}$$

Finally, F_β scores, with $\beta = \sqrt{b/1-b}$, are on the right-hand side that joins TPR to PPV, F_1 being located in the center of this side following:

$$F_1 = \frac{P(\{tp\})}{P(\{tp\}) + 0.5\,P(\{fp\}) + 0.5\,P(\{fn\})}\,. \tag{9}$$

As shown in this section, the Tile organizes an infinity of ranking scores into a 2D map through the concept of importance. Each point on the Tile corresponds to a score that assigns varying importance values to tp, tn, fp, and fn. This makes the Tile a useful visual tool for displaying the values these scores take for a given entity. Through our scenarios, we explain how to map different values on this tile for different user profiles.

Description of our illustrative context. We illustrate each scenario with a semantic segmentation example. For evaluating the performance of entities, we use the BDD100K dataset [62] as a testing set and define a two-class problem by taking the union of the first 11 semantic classes of the dataset as the negative class, corresponding to background objects (*e.g.*, road, sidewalk, or building), and the union of the 8 remaining classes as the positive class, corresponding to foreground, potentially moving, objects (*e.g.*, person, rider, or car). This type of binary split has been commonly used to improve motion detection algorithms [5, 43].

For the choice of entities, we compare and rank 74 state-of-the-art semantic segmentation models, trained on different learning sets and available in the MMSegmentation toolbox [15]. As shown in Tab. 1, we gather respectively 31 models trained on ♠ Cityscapes [16], 27 on

Table 1. List of the models (columns) sorted in chronological order based on the date of publication (from 2015 to 2023) from the MMSegmentation toolbox [15] trained on one or multiple datasets (rows). We respectively have 31 models trained on the ♠ Cityscapes [16] dataset, 27 on the ♥ ADE20K [70] dataset, 12 on the ♦ Pascal VOC 2012 [19] dataset, and 4 on the ♣ COCO-Stuff 164k [8] dataset, totaling 74 models.

	FCN [39, 50]	DeepLabV3 [11]	PSPNet [66]	ERFNet [49]	DeepLabV3+ [12]	EncNet [64]	NonLocal Net [53]	BiSeNetV1 [60]	ICNet [67]	PSANet [68]	UPerNet [56]	FastFCN [54]	Fast-SCNN [46]	ISANet [30]	APCNet [29]	DANet [24]	Semantic FPN [34]	ANN [71]	CCNet [31]	DMNet [28]	EMANet [37]	GCNet [9]	PointRend [35]	DNLNet [59]	OCRNet [63]	CGNet [55]	SETR [69]	BiSeNetV2 [61]	STDC [20]	DPT [48]	Segmenter [51]	K-Net [65]	MaskFormer [13]	SegFormer [57]	Mask2Former [14]	SAN [58]
♠ Cityscapes [16]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓						
♥ ADE20K [70]	✓	✓	✓		✓	✓	✓			✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	
♦ Pascal VOC 2012 [19]	✓	✓	✓		✓		✓				✓			✓		✓		✓	✓			✓			✓										
♣ COCO-Stuff 164k [8]		✓	✓					✓																											✓

♥ ADE20K [70], 12 on ♦ Pascal VOC 2012 [19], and 4 on ♣ COCO-Stuff 164k [8]. The classes used as learning sets are also grouped into a positive and a negative class, following the same logic as for BDD100K. Since the testing set is different from all learning sets, we are therefore evaluating the generalization capability of semantic segmentation models in a two-class setting. An overview of the results is presented through different flavors of the Tile that illustrate the scenarios. For more details, we provide a complete report on all 74 models in the supplementary material.

3.2. Four Scenarios of our Hitchhiker's Guide

In this section, we provide a step-by-step guide for using the Tile to rank two-class classifiers (*i.e.*, entities), by illustrating its various potential uses through four scenarios. Each scenario corresponds to one user profile among (1) the theoretical analyst, who is interested in understanding the theoretical relationship between different scores typically used for evaluating or ranking methods, (2) the method designer, who needs to analyze the performances of his/her new method and compare it to others, (3) the benchmarker, who organizes challenges for the scientific community and has to rank participating methods, and finally (4) the application developer, who should be able to select the most appropriate method for his/her application. In the final scenario, we also discuss various strategies for selecting an entity based on the Tile.

Scenario 1: The Theoretical Analyst

The theoretical analyst seeks to understand the foundational principles behind the scores used to evaluate and rank classifiers. Unlike other users, his/her focus is not on a specific application but rather on the theoretical relationships between various ranking scores. For this user, it is crucial to explore how different scores compare to one another, and how they correlate in terms of both value and ranking. The analyst aims to ensure that each selected score provides unique, non-redundant information, thereby enriching the evaluation process. Therefore, he/she requires a tool that can effectively illustrate the relationships between scores, helping discern whether the chosen scores are complementary or overlapping in the information they convey. To address these needs, the next section introduces the Correlation Tile, specifically designed for this type of analysis.

Correlation Tile. The Correlation Tile displays the correlation, using a linear (Pearson's r) or a rank (Spearman's ρ) correlation coefficient, between a score X, typically one used as a reference in a research field of interest, and the canonical ranking scores R_I across the Tile. The Correlation Tile is defined as:

Definition 2. *The Correlation Tile is the mapping*

$$corr_f : [0,1]^2 \to [-1,1] : (a,b) \mapsto f(X, R_I)\,,$$

where f is a correlation coefficient.

Figure 4 illustrates this correlation for the mean intersection over union (mIoU), a score commonly used to benchmark semantic segmentation models. A strong correlation between the mIoU and the Tile scores is observed in a horizontal band near the top. Hence, selecting a score in this band will lead to similar conclusions than the ones provided by the mIoU. However, selecting a score outside this band will most probably allow understanding the performances of semantic segmentation algorithms under a fresh angle. This Correlation Tile is therefore a powerful tool for understanding how to select complementary scores.

Scenario 2: The Method Designer

The method designer focuses on evaluating the performance of his/her newly developed method, comparing it against state-of-the-art and other baseline methods. This user is particularly interested in understanding how a method performs across different importances given to true positives (tp), false positives (fp), false negatives (fn), and true negatives (tn). In cases of a parametric method, the designer also seeks insights into the optimal hyper-parameter settings that maximize performance. Hereafter, we show how the Value Tile, Baseline Value Tile, and State-of-the-Art Value Tile can help the method designer understand the strengths

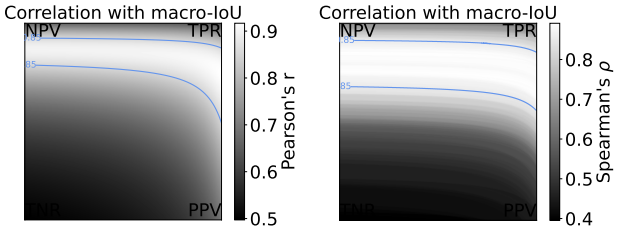


Figure 4. **Correlation Tile for linear (Pearson's r, left) and rank (Spearman's ρ, right) correlation coefficients.** These tiles show the linear and rank correlations between the canonical ranking scores and the macro-averaged IoU score that is usually taken as reference in the field of semantic segmentation, computed for the 31 models trained on ♠ Cityscapes. The blue lines delineate the area where the correlation coefficients are ≥ 0.85.

and weaknesses of a method, providing insights needed to fine-tune performance and optimize hyper-parameters.

Value Tile. The Value Tile is a map displaying the value of each ranking score R_I across the Tile for a given entity and is defined as:

Definition 3. *For an entity ϵ, the* Value Tile *is the mapping*

$$V_\epsilon : [0,1]^2 \to [0,1] : (a,b) \mapsto R_I(P_\epsilon)\,.$$

In practice, several methods can be used to compute the *Value Tile*, depending on the availability of data. (1) *Direct computation:* If the values of $P(\{fp\})$, $P(\{fn\})$, $P(\{tp\})$, and $P(\{tn\})$ are known, the first method consists in computing the values of the ranking scores in each point of the Tile using Eq. (3). (2) *Interpolation:* Knowing the values of the scores TPR, TNR, NPV, and PPV at the four corners of the Tile, the values of the remaining scores can be determined by averaging, using the f-mean: in particular, vertically, the harmonic mean $f : x \mapsto x^{-1}$ and, horizontally, an f-mean defined as $f : x \mapsto (1-x)^{-1}$. As one can see, the non-linearities differ between the vertical and horizontal axes. (3) *Equation system resolution:* The third method involves first, determining the values of $P(\{fp\})$, $P(\{fn\})$, $P(\{tp\})$, and $P(\{tn\})$ by solving a system of 4 equations with 4 unknowns, then, computing the values of the ranking scores using Eq. (3). Specifically, if the values of three ranking scores from the Value Tile are known, or if two ranking scores and the priors (for fixed priors) with $\pi_- = P(\{tn\}) + P(\{fp\})$ and $\pi_+ = P(\{fn\}) + P(\{tp\})$, then, knowing that

$$P(\{fp\}) + P(\{fn\}) + P(\{tp\}) + P(\{tn\}) = 1 \quad (10)$$

makes the system complete. Let us note that all three methods require discretizing the Tile. This can be implemented with a grid size parameter (set to 2001 in our code), which defines linearly spaced values for the two axes, a and b.

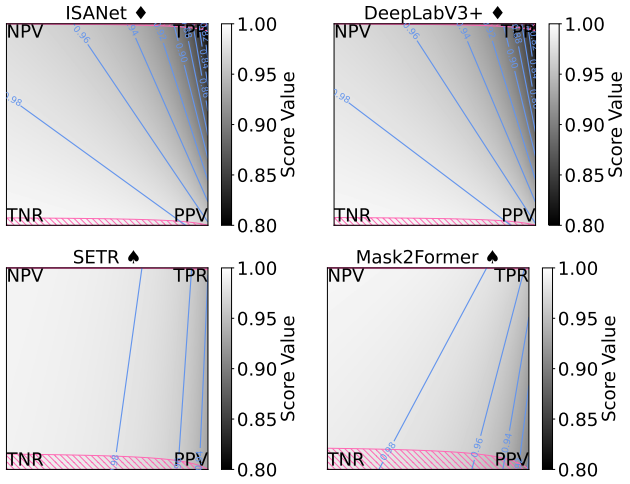


Figure 5. **Value Tile for the 4 entities ranked first (see Fig. 8).** The Value Tile shows the values of the canonical ranking scores. Blue lines are iso-value lines, *i.e.*, lines on which all scores have the same value. Hatched areas indicate regions where non-skilled performances (*i.e.*, when the ground truth Y and the prediction $\hat{Y}$ are independent) surpass that of the entity.

Figure 5 shows the Value Tile for the 4 entities ranked first among the 74 state-of-the-art semantic segmentation models (see Fig. 8). This Tile can be used to get in one look where the entity performs best, and therefore where improvements should come from the method designer.

Baseline Value Tile and State-of-the-Art Value Tile. The Baseline Value Tile is a map displaying, for a given set of entities, the minimum value for each ranking score R_I across the Tile, while the State-of-the-Art Value Tile is a map displaying the maximum value for each ranking score R_I. They are respectively defined as follows.

Definition 4. *For a given set $\mathbb{E}$ of entities, the Baseline Value Tile is the mapping*

$$[0,1]^2 \to [0,1] : (a,b) \mapsto \min_{\epsilon \in \mathbb{E}} R_I(P_\epsilon)\,.$$

Definition 5. *For a given set $\mathbb{E}$ of entities, the State-of-the-Art Value Tile is the mapping*

$$[0,1]^2 \to [0,1] : (a,b) \mapsto \max_{\epsilon \in \mathbb{E}} R_I(P_\epsilon)\,.$$

In other words, for a given set $\mathbb{E}$ of entities, the Baseline Value Tile gives the value of each score corresponding to the entity that is ranked last in each point of the Tile, while the State-of-the-Art Value Tile gives the value of the score corresponding to the entity that is ranked first.

The left-hand side of Fig. 6 illustrates the Baseline Value Tile for the 74 semantic segmentation models, showing the lowest canonical ranking score values in each point among all compared entities, while the right-hand side illustrates

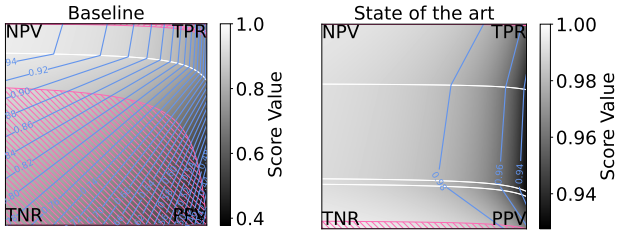


Figure 6. **Baseline Value Tile (left) and State-of-the-Art Value Tile (right).** These two tiles describe the current benchmark in semantic segmentation. The Baseline Value Tile and the State-of-the-Art Value Tile provide the *infimum* and the *supremum*, respectively, of the canonical ranking scores for the 74 semantic segmentation models. In other words, they give the value of the scores for the entities ranked last and first, respectively. The blue lines are iso-value lines, *i.e.*, lines on which all scores have the same value. The white lines mark the boundaries of these entities.

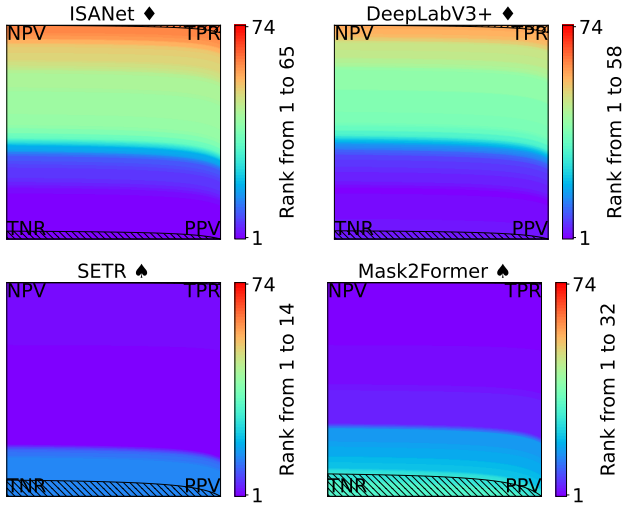


Figure 7. **Ranking Tile for the 4 entities that are ranked first, somewhere on the Tile.** The Ranking Tile shows the ranks of a given entity across the tile. Hatched areas highlight regions of the tile where no-skilled performances surpass that of the entity. Remarkably, we directly see that (1) ISANet ♦ and DeepLabV3+ ♦ have poor rankings in the upper part of the tile, while (2) SETR ♠ has the best overall performance and remains rather stable on the Tile, ranging from rank 1 to 14.

the State-of-the-Art Value Tile for the 74 semantic segmentation models, showing the highest canonical ranking score values in each point among all compared entities. The hatched areas indicate that, if the classifiers always predicting the positive class, c_+, or the negative class, c_-, were added to the set of entities to rank, they would occupy the top rank within these areas. These tiles are therefore interesting for the method designer when he/she compares them with the Value Tile of his/her entity. One can then easily compare where a method is close or above the State-of-the-Art Value Tile and where it is close or below the Baseline Value Tile. This indicates the advantages and drawbacks of a new method compared with previous works. Furthermore, these tiles may only consider a parametric family of the method, showing where different hyper-parameters perform well or bad, guiding the designer's choice.

Scenario 3: The Benchmarker

The benchmarker is focused on comparing methods from the literature to identify which ones outperform others. This user profile aims to establish a clear ranking among different methods, ensuring that their assessment is fair and consistent. Additionally, the benchmarker may be interested in organizing open challenges, where the goal is to evaluate and rank participating methods to determine the top performer. In this context, it becomes crucial to accurately determine a winner by ranking the different entries based on their performance. To meet these needs, the benchmarker requires a tool that can generate reliable rankings.

Ranking Tile. The Ranking Tile maps, for a given entity ϵ, the rank of that entity across the tile. For a given set of entities $\mathbb{E}$, rankings are based on an ordering of the performances, defined in paper A [44], induced by the ranking scores R_I, denoted by $\lesssim_{R_I}$. The Ranking Tile is defined as:

Definition 6. *The Ranking Tile is the mapping*

$$[0,1]^2 \to [1, |\mathbb{E}|] : (a,b) \mapsto \mathrm{rank}_{\mathbb{E}}(\epsilon)\,,$$

where $\mathrm{rank}_{\mathbb{E}}(\epsilon)$ *is computed according to* $\lesssim_{R_I}$.

Therefore, the Ranking Tile can be obtained by ordering the performances for each ranking score, *i.e.*, ordering in each point (a,b) the values of the ranking score for the Value Tile of each entity of the set. This means that the Ranking Tile is discretized similarly to the Value Tile.

Figure 7 shows the Ranking Tile for the 4 entities ranked first among the 74 state-of-the-art semantic segmentation models (see Fig. 8). The Value Tile (Fig. 5) and the Ranking Tile (Fig. 7) provide, at a glance, an overview of an entity's performance across the entire tile.

The Ranking Tile is therefore a great solution to help the benchmarker identify the best-performing methods and appropriately determine the winners in open challenges, as it provides a structured way to know in which case a method has a particular rank.

Scenario 4: The Application Developer

The application developer is interested in selecting the best method that aligns with the specific requirements of his/her

application. This developer may already know the importance of true positives, false positives, false negatives, and true negatives relevant to a particular use case. The goal is therefore to choose a method that best meets these predefined criteria. For this purpose, the application developer needs a tool that allows him/her to efficiently identify the most suitable method based on specific priorities.

Entity Tile. The Entity Tile maps entities that are at a given rank across the tile and is defined as follows.

Definition 7. *The Entity Tile, for a given rank* $r \in [1, |\mathbb{E}|]$*, is the mapping*

$$[0,1]^2 \to \mathbb{E} : (a,b) \mapsto \epsilon_r \,,$$

where ϵ_r *is the entity ranked* r*-th, according to* $\lesssim_{R_I}$*.*

The Entity Tile thus shows all entities that are at a given rank and is constructed similarly to the Ranking Tile. In the special case of $r = 1$, the Entity Tile shows the best entities among $\mathbb{E}$, while for $r = |\mathbb{E}|$, it shows the worst entities among $\mathbb{E}$. As illustrated in Fig. 8 for the semantic segmentation models, the first rank is shared by 4 entities, namely Mask2Former and SETR, both trained on ♠ Cityscapes, and DeepLabV3+ and ISANet, both trained on ♦ Pascal VOC 2012. These 4 entities also appear in the second and third ranks in other areas of the ranking tiles, *i.e.*, for other canonical ranking scores. Interestingly, the properties ensure that entities ranked first are the ones on the upper dashed broken line of Fig. 2 in the ROC space, and the ones ranked last are on the lower dashed broken line [2].

The application developer may choose an entity based on those tiles depending on 3 cases, based on the knowledge of importance values: (1) In the first case, the importance values are *known*. Hence, the application developer simply selects the method ranked number 1 at the corresponding place on the Tile. This is typically how most benchmarks rank methods, *i.e.*, based on a single score. (2) In the second case, the importance values are *unknown but can be determined* by analyzing the community practices. For example, in semantic segmentation, the "mean intersection over union" (mIoU) is often considered to be a good criterion. Although this score is not part of the Tile, the Correlation Tile provides a direct comparison between the orderings induced by all ranking scores and the macro-averaged IoU. The application developer can then simply select the method ranked first in the area, where Spearman's ρ is the highest in the tile. (3) In the last case, there is *no information* on the importance values. One selection mechanism consists in minimizing the maximum rank over the Ranking Tile and, in case of ex aequo, minimizing the average rank.

Therefore, the Entity Tile enables the application developer to select the best method by visualizing how different options perform relative to the importance values, whether they are known or not. This allows to make informed decisions tailored to specific application requirements.

[2]Further details on the link between the Tile and ROC are provided in paper B [45].

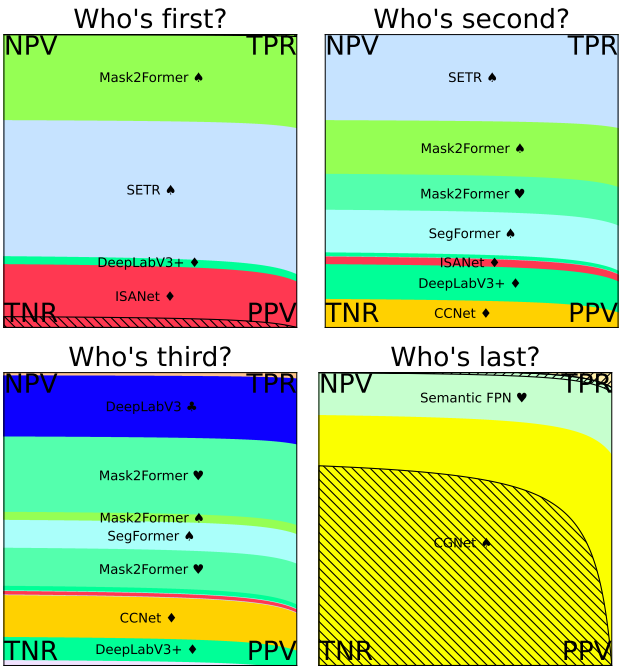


Figure 8. **Entity Tile showing the entities that are ranked at the first, second, third, and last positions.** Hatched areas in the "who's first?" and "who's last?" tiles highlight regions where no-skilled performances surpass that of the entities in the current map.

4. Conclusion

In this paper, we introduced a practical hitchhiker's guide to understanding the performance of two-class classifiers. We organized our guide into four distinct scenarios, each corresponding to a specific user profile, covering the theoretical analyst, the method designer, the benchmarker, and the application developer. For each scenario, we provided practical examples of challenges each user may encounter when evaluating classifier performance and identified the most suitable tools for their needs. Central to our approach is the adaptation of the Tile, which arranges an infinite number of ranking scores into a 2D map. By leveraging different mappings, we demonstrated the versatility of the Tile and its various flavors to address the diverse requirements of these user profiles. This guide offers a flexible and robust framework that enables users to effectively evaluate, rank, and interpret the performance of two-class classifiers, making it a valuable resource for researchers, practitioners, and developers alike.

Acknowledgments. The work by A. Halin and S.

Piérard was supported by the Walloon Region (Service Public de Wallonie Recherche, Belgium) under grant n°2010235 (ARIAC by DIGITALWALLONIA4.AI). A. Cioppa is funded by the F.R.S.-FNRS.

References

[1] Azim Ahmadzadeh, Dustin J. Kempton, Petrus C. Martens, and Rafal A. Angryk. Contingency space: A semimetric space for classification evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):1501–1513, 2023. 2

[2] Donald Bamber. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J. Math. Psychol.*, 12(4):387–415, 1975. 2

[3] Ricardo Barandela, Josep Salvador Sánchez, Vicente García, and Erick Rangel. Strategies for learning in class imbalance problems. *Pattern Recognit.*, 36(3):849–851, 2003. 2

[4] Kendrick Boyd, Victor Costa, Jesse Davis, and David Page. Unachievable region in precision-recall space and its effect on empirical evaluation. In *Int. Conf. Mach. Learn. (ICML)*, pages 639–646, Edinburgh, UK, 2012. 2

[5] Marc Braham, Sébastien Piérard, and Marc Van Droogenbroeck. Semantic background subtraction. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 4552–4556, Beijing, China, 2017. 4

[6] William M. Briggs and Russell Zaretzki. The skill plot: A graphical technique for evaluating continuous diagnostic tests. *Biometrics*, 64(1):250–256, 2008. 2

[7] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *IEEE Int. Conf. Pattern Recognit. (ICPR)*, pages 3121–3124, Istanbul, Turkey, 2010. 2

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-stuff: Thing and stuff classes in context. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1209–1218, Salt Lake City, UT, USA, 2018. 5, 13

[9] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. GCNet: Non-local networks meet squeeze-excitation networks and beyond. In *IEEE/CVF Int. Conf. Comput. Vis. Work. (ICCV Work.)*, pages 1971–1980, Seoul, South Korea, 2019. 5

[10] Andre M. Carrington, Douglas G. Manuel, Paul W. Fieguth, Tim Ramsay, Venet Osmani, Bernhard Wernly, Carol Bennett, Steven Hawken, Olivia Magwood, Yusuf Sheikh, Matthew McInnes, and Andreas Holzinger. Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):329–341, 2023. 2

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv*, abs/1706.05587, 2017. 5

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 833–851, Munich, Germany, 2018. 5

[13] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, pages 17864–17875. Curran Assoc. Inc., 2021. 5

[14] 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)*, pages 1280–1289, New Orleans, LA, USA, 2022. 5

[15] MMSegmentation Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 4, 5

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3213–3223, Las Vegas, NV, USA, 2016. 4, 5, 13

[17] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Int. Conf. Mach. Learn. (ICML)*, pages 233–240, Pittsburgh, Pennsylvania, 2006. 2

[18] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006. 2

[19] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 5, 13

[20] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking BiSeNet for real-time semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 9711–9720, Nashville, TN, USA, 2021. 5

[21] Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. *Mach. Learn.*, 31:1–38, 2004. 2

[22] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognit. Lett.*, 27(8):861–874, 2006. 2

[23] Peter Flach. The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In *Int. Conf. Mach. Learn. (ICML)*, pages 194–201, Washington, DC, USA, 2003. 2

[24] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3141–3149, Long Beach, CA, USA, 2019. 5

[25] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *Int. Conf. Nat. Comput.*, pages 192–201, Jinan, China, 2008. 2

[26] Anaïs Halin, Sébastien Piérard, Anthony Cioppa, and Marc Van Droogenbroeck. A hitchhiker's guide to understanding performances of two-class classifiers. *arXiv*, abs/2412.04377, 2024. 1

[27] Frank E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer International Publishing, 2015. 2

[28] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 3561–3571, Seoul, South Korea, 2019. 5

[29] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7511–7520, Long Beach, CA, USA, 2019. 5

[30] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv*, abs/1907.12273, 2019. 5

[31] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 603–612, Seoul, South Korea, 2019. 5

[32] Jacqueline M. Hughes-Oliver. Population and empirical PR curves for assessment of ranking algorithms. *arXiv*, abs/1810.08635, 2018. 2

[33] Nathalie Japkowicz and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Camb. Univ. Press, 2011. 2

[34] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 9396–9405, Long Beach, CA, USA, 2019. 5

[35] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 9796–9805, Seattle, WA, USA, 2020. 5

[36] Camelia Lemnaru and Rodica Potolea. Imbalanced classification problems: systematic study, issues and best practices. In *Enterp. Inf. Syst.*, pages 35–50. Springer Berl. Heidelb., 2012. 2

[37] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 9166–9175, Seoul, South Korea, 2019. 5

[38] Yusha Liu, Yichong Xu, Nihar B. Shah, and Aarti Singh. Integrating rankings into quantized scores in peer review. *Trans. Mach. Learn. Res.*, pages 1–28, 2022. 2

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3431–3440, Boston, MA, USA, 2015. 5

[40] Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *Int. Conf. Mach. Learn. (ICML)*, pages 603–611, Atlanta, Georgia, USA, 2013. 3

[41] Francis Sahngun Nahm. Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 75(1):25–36, 2022. 2

[42] Tran Thien Dat Nguyen, Hamid Rezatofighi, Ba-Ngu Vo, Ba-Tuong Vo, Silvio Savarese, and Ian Reid. How trustworthy are performance evaluations for basic vision tasks? *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(7):8538–8552, 2023. 3

[43] Hyeoncheol Noh, Jingi Ju, Minseok Seo, Jongchan Park, and Dong-Geol Choi. Unsupervised change detection based on image reconstruction loss. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. (CVPRW)*, pages 1351–1360, New Orleans, LA, USA, 2022. 4

[44] Sébastien Piérard, Anaïs Halin, Anthony Cioppa, Adrien Deliège, and Marc Van Droogenbroeck. Foundations of the theory of performance-based ranking. *arXiv*, abs/2412.04227, 2024. 1, 2, 3, 7

[45] 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*, abs/2412.04309, 2024. 1, 2, 3, 8, 12

[46] Rudra P. K. Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-SCNN: Fast semantic segmentation network. *arXiv*, abs/1902.04502, 2019. 5

[47] David Martin Ward Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol.*, 2(1):37–63, 2011. 2

[48] Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 12159–12168, Montréal, Can., 2021. 5

[49] Eduardo Romera, Jose M. Alvarez, Luis M. Bergasa, and Roberto Arroyo. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst.*, 19(1):263–272, 2018. 5

[50] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017. 5

[51] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 7242–7252, Montréal, Can., 2021. 5

[52] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2018. 2

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7794–7803, Salt Lake City, UT, USA, 2018. 5

[54] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv*, abs/1903.11816, 2019. 5

[55] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yongdong Zhang. CGNet: A light-weight context guided network for semantic segmentation. *IEEE Trans. Image Process.*, 30: 1169–1179, 2021. 5

[56] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 432–448. Springer Int. Publ., 2018. 5

[57] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers.

In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, pages 12077–12090, 2021. 5

[58] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2945–2954, New Orleans, LA, USA, 2023. 5

[59] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 191–207. Springer Int. Publ., 2020. 5

[60] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 325–341. Springer Int. Publ., 2018. 5

[61] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.*, 129(11):3051–3068, 2021. 5

[62] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2633–2642, Seattle, WA, USA, 2020. 4, 13

[63] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 173–190. Springer Int. Publ., 2020. 5

[64] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7151–7160, Salt Lake City, UT, USA, 2018. 5

[65] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, pages 10326–10338. Curran Assoc. Inc., 2021. 5

[66] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6230–6239, Honolulu, HI, USA, 2017. 5

[67] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. ICNet for real-time semantic segmentation on high-resolution images. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 418–434. Springer Int. Publ., 2018. 5

[68] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 270–286. Springer Int. Publ., 2018. 5

[69] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6877–6886, Nashville, TN, USA, 2021. 5

[70] 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)*, pages 5122–5130, Honolulu, HI, USA, 2017. 5, 13

[71] Zhen Zhu, Mengdu Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 593–602, Seoul, South Korea, 2019. 5

A. Supplementary Material

We provide in this supplementary material (1) the list of symbols used in this paper (Appendix A.1), (2) a description of the software in the Python Jupyter Notebook (Appendix A.2), (3) a definition of the Relative-Skill Tile (Appendix A.3), (4) more details about the illustration used in the paper (Appendix A.4), (5) an analysis of the behavior of scores, using the Correlation Tile, explaining the patterns observed in the various tiles presented in the paper (Appendix A.5), and finally (6) the comprehensive report generated by the Python Jupyter Notebook for our illustration (Appendix A.6).

A.1. List of Symbols

Symbols used for the probability theory

- Ω: the sample space (universe)
- Σ: the event space (a σ-algebra on Ω, *e.g.* 2^{Ω})
- (Ω, Σ): the measurable space

Symbols used for our probabilistic framework for performances

- P: a performance, *i.e.*, a probability measure
- P_{ϵ}: the performance of an entity ϵ
- $\mathbb{P}_{(\Omega,\Sigma)}$: all performances on (Ω, Σ)
- X: a score
- $\mathbb{X}_{(\Omega,\Sigma)}$: all scores on (Ω, Σ)
- $\mathrm{dom}(X)$: the domain of the score X

Symbols used for two-class classifications

- Y: the random variable for the ground truth
- $\hat{Y}$: the random variable for the prediction
- c_{-}: the negative class
- c_{+}: the positive class
- tn: the sample *true negative*
- fp: the sample *false positive*
- fn: the sample *false negative*
- tp: the sample *true positive*
- A: the score *accuracy*
- TNR: the score *true negative rate*
- TPR: the score *true positive rate*
- FPR: the score *false positive rate*
- NPV: the score *negative predictive value*
- PPV: the score *positive predictive value*
- F_{β}: the F-scores
- π_{+}: the score *prior of the positive class*
- π_{-}: the score *prior of the negative class*
- ROC: Receiver Operating Characteristic
- PR: Precision-Recall
- AUC-ROC: Area Under The ROC curve
- AUC-PR: Area Under The PR curve

Symbols used for the performance-based ranking of entities

- $\mathrm{rank}_{\mathbb{E}}$: the *ranking* function, relative to the set of entities $\mathbb{E}$
- $\mathbb{E}$: the set of entities to rank
- ϵ: an entity (*i.e.*, an element of $\mathbb{E}$)
- I: the random variable *Importance*
- R_I: the *ranking score* parameterized by the importance I
- $\lesssim_{R_I}$: the ordering induced by the ranking score R_I
- a: the parameter specifying the relative importance given to the incorrect outcomes (*i.e.*, tp and tn), it corresponds to the horizontal axis of the Tile
- b: the parameter specifying the relative importance given to the correct outcomes (*i.e.*, fn and fp), it corresponds to the vertical axis of the Tile

Symbols used for our illustration

- ♠: the learning set *Cityscapes*
- ♥: the learning set *ADE20K*
- ♦: the learning set *Pascal VOC 2012*
- ♣: the learning set *COCO-Stuff 164k*

Other symbols

- $\mathbb{R}$: the real numbers
- mIoU: mean intersection over union
- r: the linear correlation coefficient of Pearson
- ρ: the rank correlation coefficient of Spearman
- τ: the rank correlation coefficient of Kendall

A.2. Software Description

The *Python Jupyter Notebook* generates a comprehensive report related to the illustration of this paper, featuring all the tiles presented in the paper and more. The various tiles are obtained by integrating two types of information. The first involves point-specific data, such as the value of a ranking score, a rank, an entity, or a correlation value, as detailed in this paper. This is achieved by discretizing the Tile using a grid size parameter along the two axis, a and b, and mapping the respective data, accordingly to the ranking scores in the Tile, to a point (a, b). The second type, relevant when priors are fixed (*i.e.*, when all compared performances have the same class priors), pertains to area-based information. The algorithm used to identify these areas of interest, described in paper B [45], determines the areas (and their respective boundaries) within the Tile where an entity holds a specific rank (typically, the first or the last) without requiring a discretization of the Tile. The hatched areas on the various tiles provided in the paper (*e.g.*, the Value Tile or Ranking Tile) are drawn using this algorithm, as well as the white boundary lines on the Baseline Value Tile and the State-of-the-Art Value Tile.

A.3. Relative-Skill Tile

In this section, we define two new tiles, namely the No-Skill Tile and the Relative-Skill Tile. For this purpose, we first define two sets of performances.

We say that a performance P is *no-skilled* when the ground truth Y and the prediction $\hat{Y}$ are independent. We denote the set of all non-skilled performances on (Ω, Σ) by $\mathbb{P}^{Y \perp\!\!\!\perp \hat{Y}}_{(\Omega,\Sigma)}$:

$$\mathbb{P}^{Y \perp\!\!\!\perp \hat{Y}}_{(\Omega,\Sigma)} = \left\{ P \in \mathbb{P}_{(\Omega,\Sigma)} : P(Y, \hat{Y}) = P(Y)P(\hat{Y}) \right\} . \quad (11)$$

Therefore, the classifier always predicting the positive class c_+ and the classifier always predicting the negative class c_- have both no-skilled performances.

When the priors are the same for all compared performances, we can define the set of all performances at a given positive prior, π_+:

$$\mathbb{P}^{\pi_+}_{(\Omega,\Sigma)} = \left\{ P \in \mathbb{P}_{(\Omega,\Sigma)} : P(Y = c_+) = \pi_+ \right\} . \quad (12)$$

Definition 8. *The* No-Skill Tile *is the mapping*

$$noskill_I : [0,1]^2 \to [0,1] : (a,b) \mapsto \max_{P \in \mathbb{P}^{Y \perp\!\!\!\perp \hat{Y}}_{(\Omega,\Sigma)} \cap \mathbb{P}^{\pi_+}_{(\Omega,\Sigma)}} R_I(P) .$$

The No-Skill Tile thus displays the value of the ranking scores in each point of the tile for the best-performing non-skilled performance. Note that, *e.g.*, the hatched areas on Fig. 5 highlight regions where the No-Skill Tile exhibits a higher value than the Value Tile.

Definition 9. *The* Relative-Skill Tile *is the mapping*

$$skill_I : [0,1]^2 \to [0,1] : (a,b) \mapsto \frac{SOTA_I - noskill_I}{1 - noskill_I} ,$$

where $SOTA_I$ corresponds to the values of the State-of-the-Art Value Tile.

Figure 9 illustrates the No-Skill Tile and the Relative-Skill Tile. Note that the code to obtain these tiles is also available in the Python Jupyter Notebook.

A.4. More Details on the Illustration

Our illustration is for an arbitrarily chosen pixel-based two-class classification task derived from a pixel-based semantic segmentation task. The performances are those of 74 models evaluated on 8,000 images of the BDD100K dataset [62]: this is our testing set. These models have been trained on the ♠ Cityscapes [16], ♥ ADE20K [70], ♦ Pascal VOC 2012 [19], or ♣ COCO-Stuff 164k [8] datasets: these are the learning sets. The set of semantic labels in BDD100K are identical to those in ♠ Cityscapes, but differ from the ones in ♥ ADE20K, ♦ Pascal VOC 2012, and ♣ COCO-Stuff 164k.

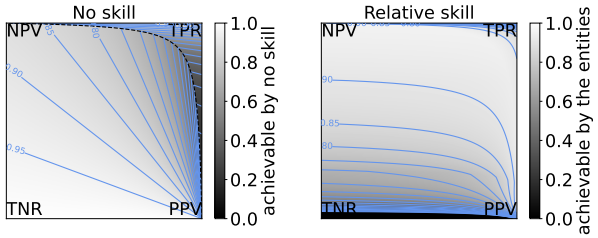


Figure 9. **No-Skill Tile (left) and Relative-Skill Tile (right).** These tiles show, on the left, the score values of the best no-skilled performances and, on the right, the relative performance of the state of the art compared to no-skilled performances for the 74 semantic segmentation models. The blue lines are iso-value lines, *i.e.*, lines on which all scores have the same value. The dashed line indicates boundaries between different entities.

Defining the two classes for the testing set. We arbitrarily took the union of the first 11 semantic labels of BDD100K as the negative class, corresponding to background objects (*e.g.*, road, sidewalk, or building), and the union of the 8 remaining ones as the positive class (person, rider, car, truck, bus, train, motorcycle, and bicycle).

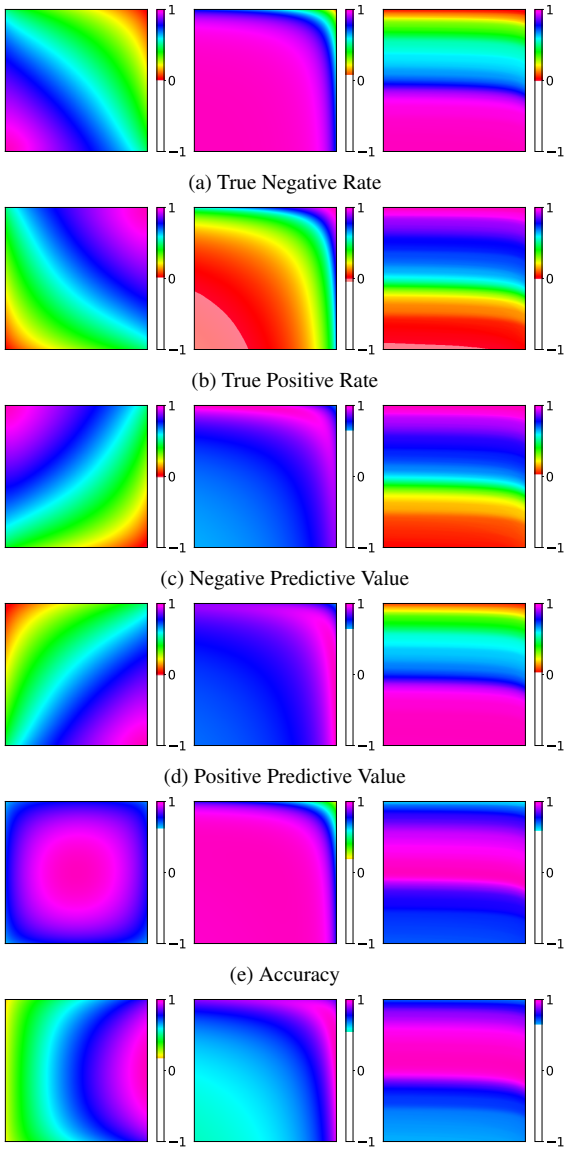
Defining the two classes for the 4 learning sets. The semantic labels predicted by the models are those from the corresponding learning set. We therefore also had to map all these labels to our negative and positive classes. To this end, for each learning set, taking into account all the models learned on this learning set, we computed the proportion of pixels that are positive for each semantic label. The semantic label has been attributed to the positive class when this proportion is greater than the positive prior π_+ of BDD100K (about 0.1242).

All the results presented in our paper, and also hereafter, are specific to these arbitrary choices. In particular, we noticed that thresholding the posteriors at 0.5 instead of π_+ leads to a significantly different two-class classification problem, for which the ranking of the models is different.

A.5. Behavior of Scores

The Correlation Tile allows to depict the behavior of any score, showing the rank correlations between that score and all canonical ranking scores, for a given performance distribution. On Fig. 10, we analyze the rank correlation, using Spearman correlation coefficient (Spearman's ρ), for 6 scores belonging to the ranking scores (namely TNR, TPR, NPV, PPV, A, and F_1) and 3 distributions of performances (all 3 distributions are uniform but are on different sets of performances). We analyze these scores (1) for a uniform distribution over all performances, (2) for a uniform distribution of performances having fixed priors corre-

sponding to those of BDD100K, and (3) for a uniform distribution of performances, where the set of performances is the one of our illustration. Comparing the figures on the left with those on the center, we see the effect of having fixed priors with $\pi_+ = 0.1242$. The figures on the right show the effect of having fixed priors and an arbitrary choice of performances (the ones of the 74 state-of-the-art semantic segmentation models). These last two conditions seems to be responsible for the patterns (horizontal bands) observed in the various tiles of the paper.



(a) True Negative Rate

(b) True Positive Rate

(c) Negative Predictive Value

(d) Positive Predictive Value

(e) Accuracy

(f) F-one

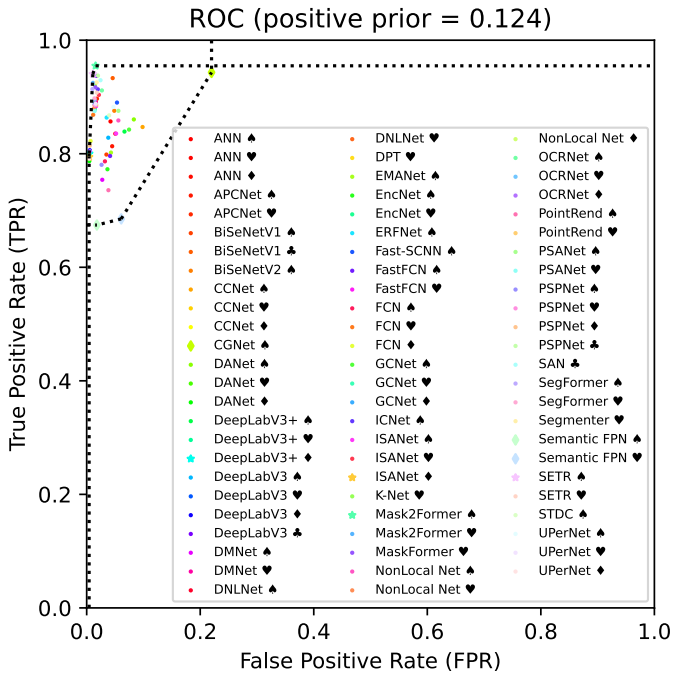
Figure 10. **Behavior of** 6 **scores for** 3 **sets of performances.** The Correlation Tile shows the estimated rank correlations (with the Spearman's ρ) between 6 scores and all canonical ranking scores, for a uniform distribution over all possible performances (left), over all performances with a prior of the positive class equal to the one in our illustration ($\pi_+ = 0.124227$) (center), and over the 74 performances compared in our illustration (right).

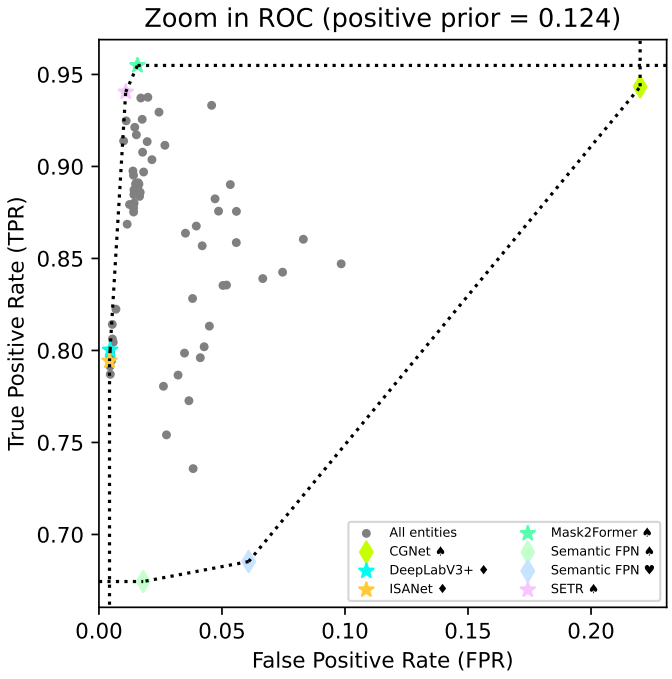
A.6. Report Generated by the Python Jupyter Notebook for our Illustration

The performances of the two-class classification entities

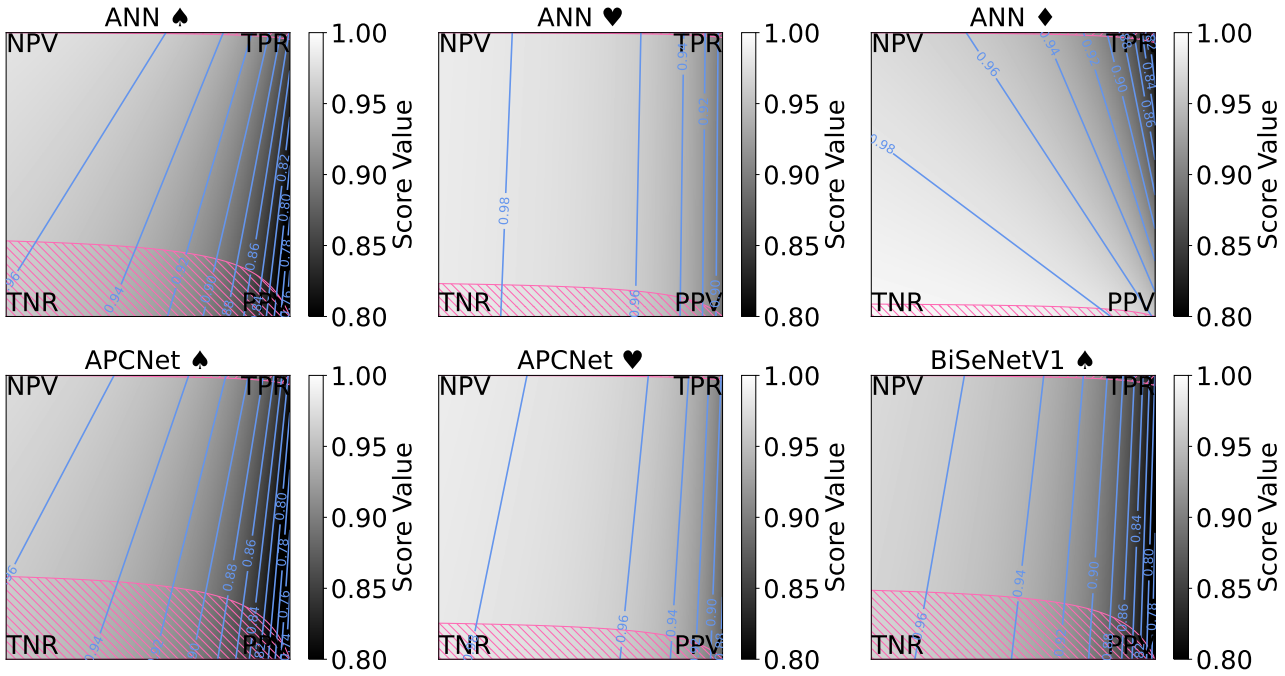
entity	$P(\{tn\})$	$P(\{fp\})$	$P(\{fn\})$	$P(\{tp\})$
ANN ♠	0.8390	0.0368	0.0368	0.1064
ANN ♥	0.8615	0.0142	0.0142	0.1106
ANN ♦	0.8713	0.0045	0.0045	0.0987
APCNet ♠	0.8365	0.0393	0.0393	0.1010
APCNet ♥	0.8599	0.0159	0.0159	0.1114
BiSeNetV1 ♠	0.8453	0.0304	0.0304	0.0992
BiSeNetV1 ♣	0.8357	0.0401	0.0401	0.1159
BiSeNetV2 ♠	0.8332	0.0426	0.0426	0.1088
CCNet ♠	0.7896	0.0862	0.0862	0.1052
CCNet ♥	0.8621	0.0136	0.0136	0.1101
CCNet ♦	0.8719	0.0039	0.0039	0.0990
CGNet ♠	0.6831	0.1926	0.1926	0.1172
DANet ♠	0.8030	0.0728	0.0728	0.1069
DANet ♥	0.8626	0.0131	0.0131	0.1107
DANet ♦	0.8717	0.0040	0.0040	0.0978
DeepLabV3+ ♠	0.8175	0.0583	0.0583	0.1042
DeepLabV3+ ♥	0.8627	0.0131	0.0131	0.1106
DeepLabV3+ ♦	0.8718	0.0040	0.0040	0.0994
DeepLabV3 ♠	0.8424	0.0333	0.0333	0.1029
DeepLabV3 ♥	0.8633	0.0125	0.0125	0.1102
DeepLabV3 ♦	0.8710	0.0048	0.0048	0.1002
DeepLabV3 ♣	0.8609	0.0149	0.0149	0.1164
DMNet ♠	0.8517	0.0240	0.0240	0.0937
DMNet ♥	0.8618	0.0140	0.0140	0.1107
DNLNet ♠	0.8316	0.0441	0.0441	0.1038
DNLNet ♥	0.8633	0.0124	0.0124	0.1099
DPT ♥	0.8635	0.0123	0.0123	0.1092
EMANet ♠	0.8383	0.0375	0.0375	0.0996
EncNet ♠	0.8437	0.0320	0.0320	0.0960
EncNet ♥	0.8633	0.0125	0.0125	0.1093
ERFNet ♠	0.8450	0.0308	0.0308	0.1073
Fast-SCNN ♠	0.8290	0.0468	0.0468	0.1106
FastFCN ♠	0.8397	0.0361	0.0361	0.0989
FastFCN ♥	0.8613	0.0144	0.0144	0.1098
FCN ♠	0.8476	0.0282	0.0282	0.0977
FCN ♥	0.8657	0.0100	0.0100	0.1079
FCN ♦	0.8697	0.0061	0.0061	0.1022
GCNet ♠	0.8104	0.0654	0.0654	0.1047
GCNet ♥	0.8635	0.0123	0.0123	0.1090
GCNet ♦	0.8715	0.0043	0.0043	0.0989
ICNet ♠	0.8586	0.0172	0.0172	0.1135
ISANet ♠	0.8304	0.0454	0.0454	0.1038
ISANet ♥	0.8569	0.0189	0.0189	0.1123
ISANet ♦	0.8720	0.0038	0.0038	0.0987
K-Net ♥	0.8630	0.0127	0.0127	0.1144
Mask2Former ♠	0.8621	0.0137	0.0137	0.1186
Mask2Former ♥	0.8661	0.0097	0.0097	0.1149
MaskFormer ♥	0.8624	0.0133	0.0133	0.1139

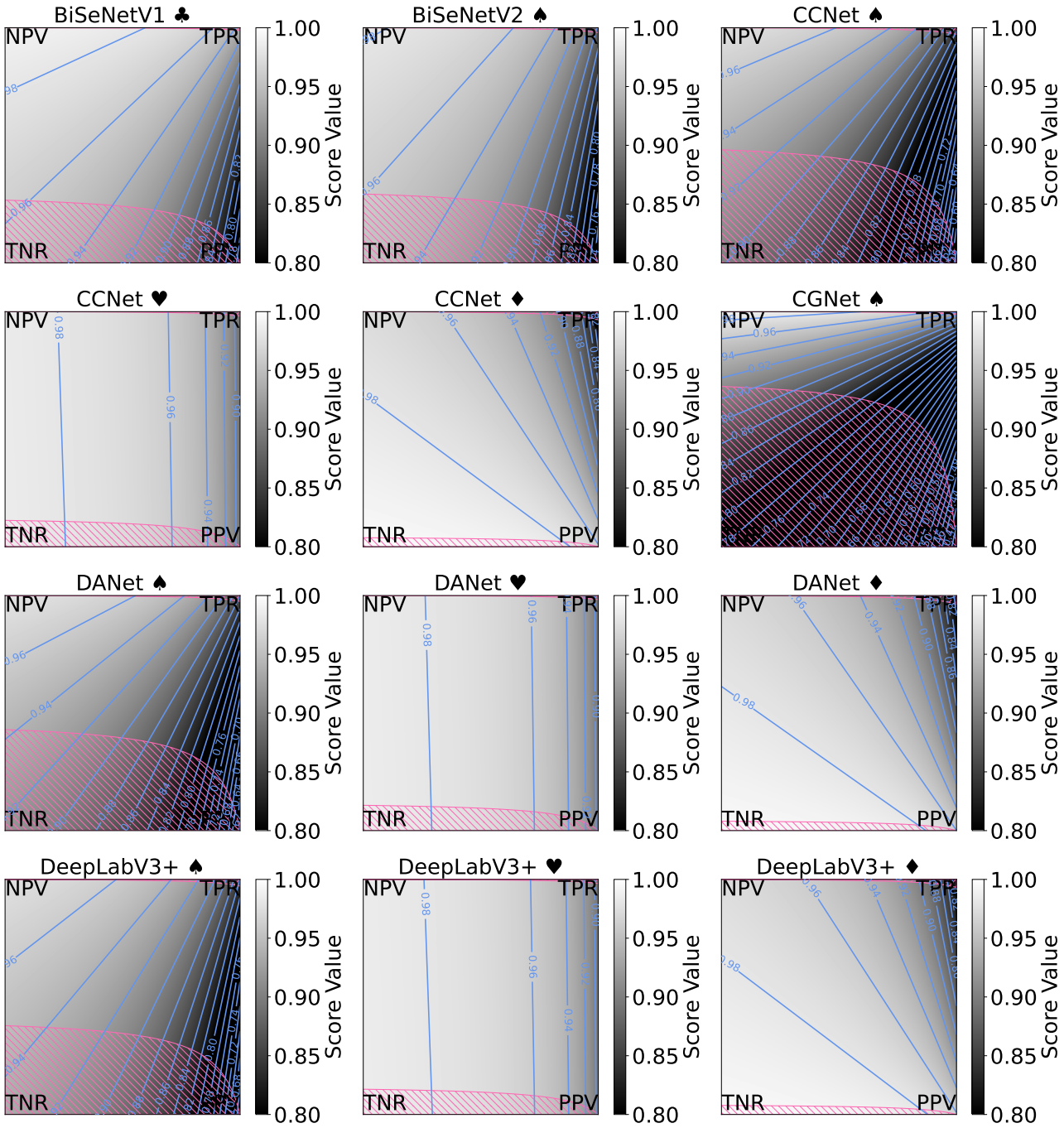
NonLocal Net ♠	0.8269	0.0489	0.0489	0.1067
NonLocal Net ♥	0.8649	0.0109	0.0109	0.1092
NonLocal Net ♦	0.8716	0.0042	0.0042	0.0992
OCRNet ♠	0.8523	0.0235	0.0235	0.1132
OCRNet ♥	0.8635	0.0123	0.0123	0.1112
OCRNet ♦	0.8707	0.0051	0.0051	0.0999
PointRend ♠	0.8422	0.0335	0.0335	0.0914
PointRend ♥	0.8611	0.0147	0.0147	0.1101
PSANet ♠	0.8268	0.0489	0.0489	0.1088
PSANet ♥	0.8634	0.0123	0.0123	0.1087
PSPNet ♠	0.8528	0.0230	0.0230	0.0970
PSPNet ♥	0.8629	0.0129	0.0129	0.1100
PSPNet ♦	0.8710	0.0048	0.0048	0.1011
PSPNet ♣	0.8584	0.0174	0.0174	0.1165
SAN ♣	0.8544	0.0214	0.0214	0.1155
SegFormer ♠	0.8670	0.0088	0.0088	0.1135
SegFormer ♥	0.8637	0.0121	0.0121	0.1115
Segmenter ♥	0.8604	0.0154	0.0154	0.1150
Semantic FPN ♠	0.8600	0.0158	0.0158	0.0838
Semantic FPN ♥	0.8225	0.0533	0.0533	0.0851
SETR ♠	0.8663	0.0095	0.0095	0.1168
SETR ♥	0.8602	0.0155	0.0155	0.1128
STDC ♠	0.8411	0.0347	0.0347	0.1078
UPerNet ♠	0.8344	0.0413	0.0413	0.1096
UPerNet ♥	0.8619	0.0138	0.0138	0.1103
UPerNet ♦	0.8718	0.0039	0.0039	0.0983

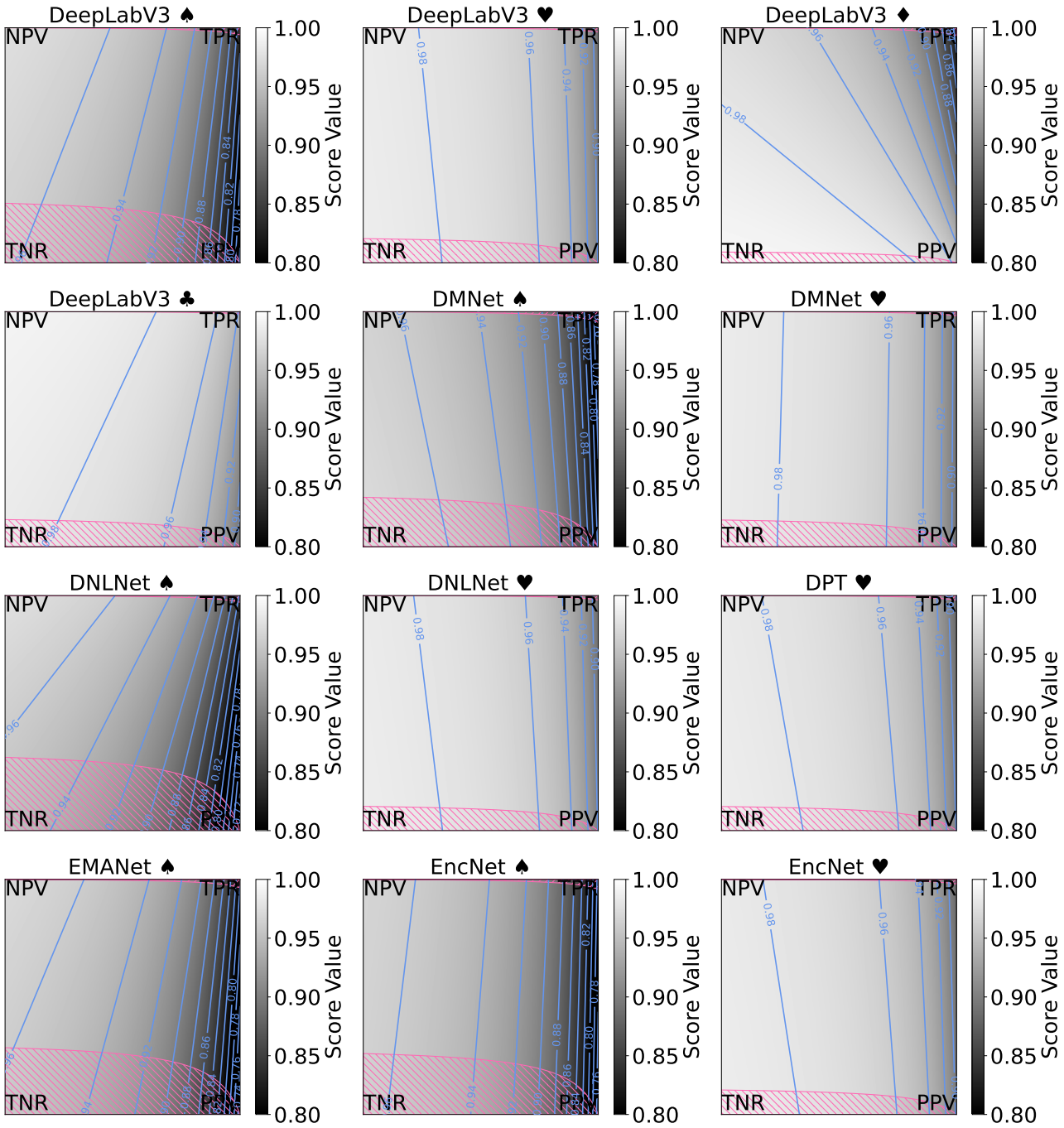


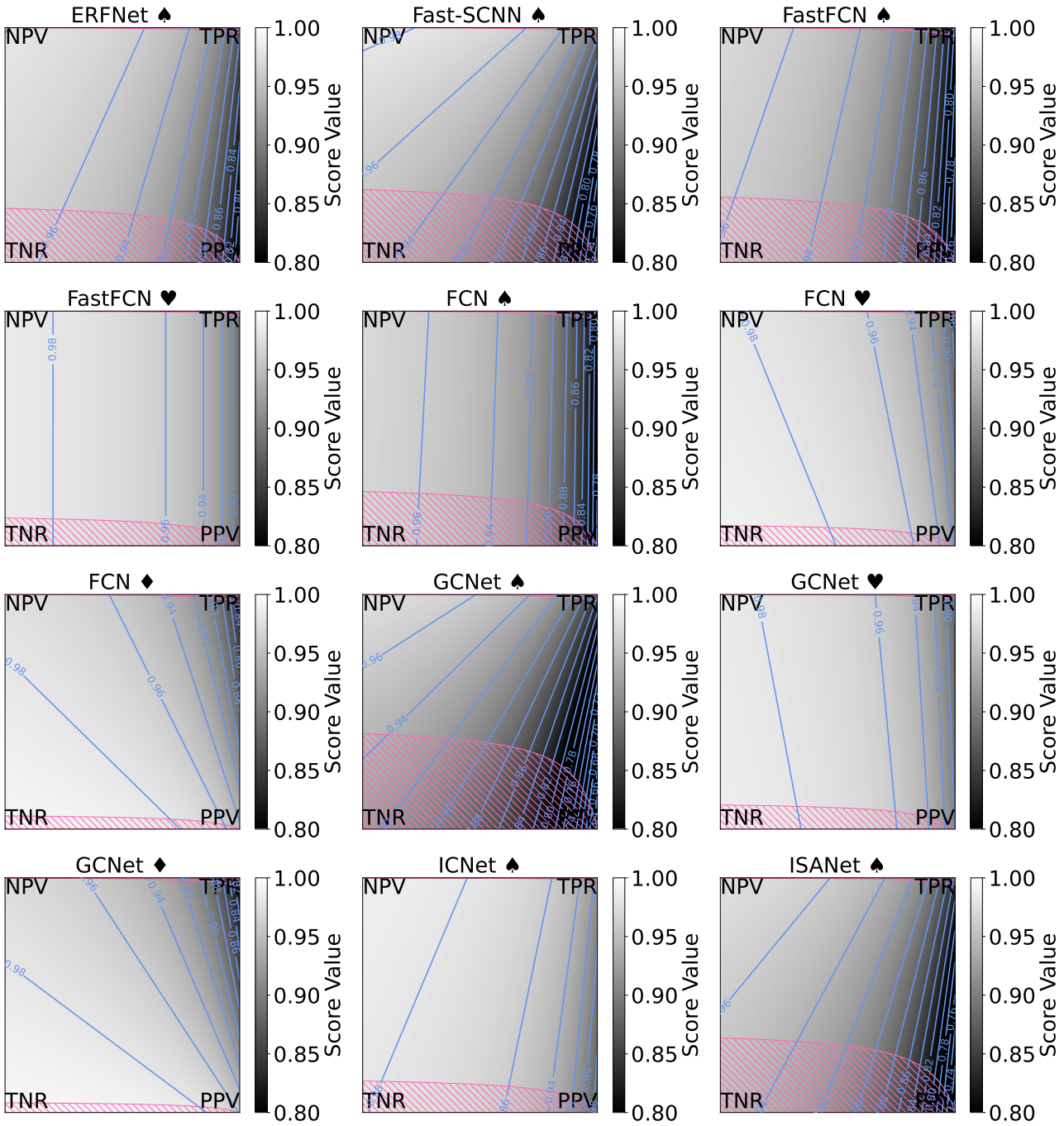


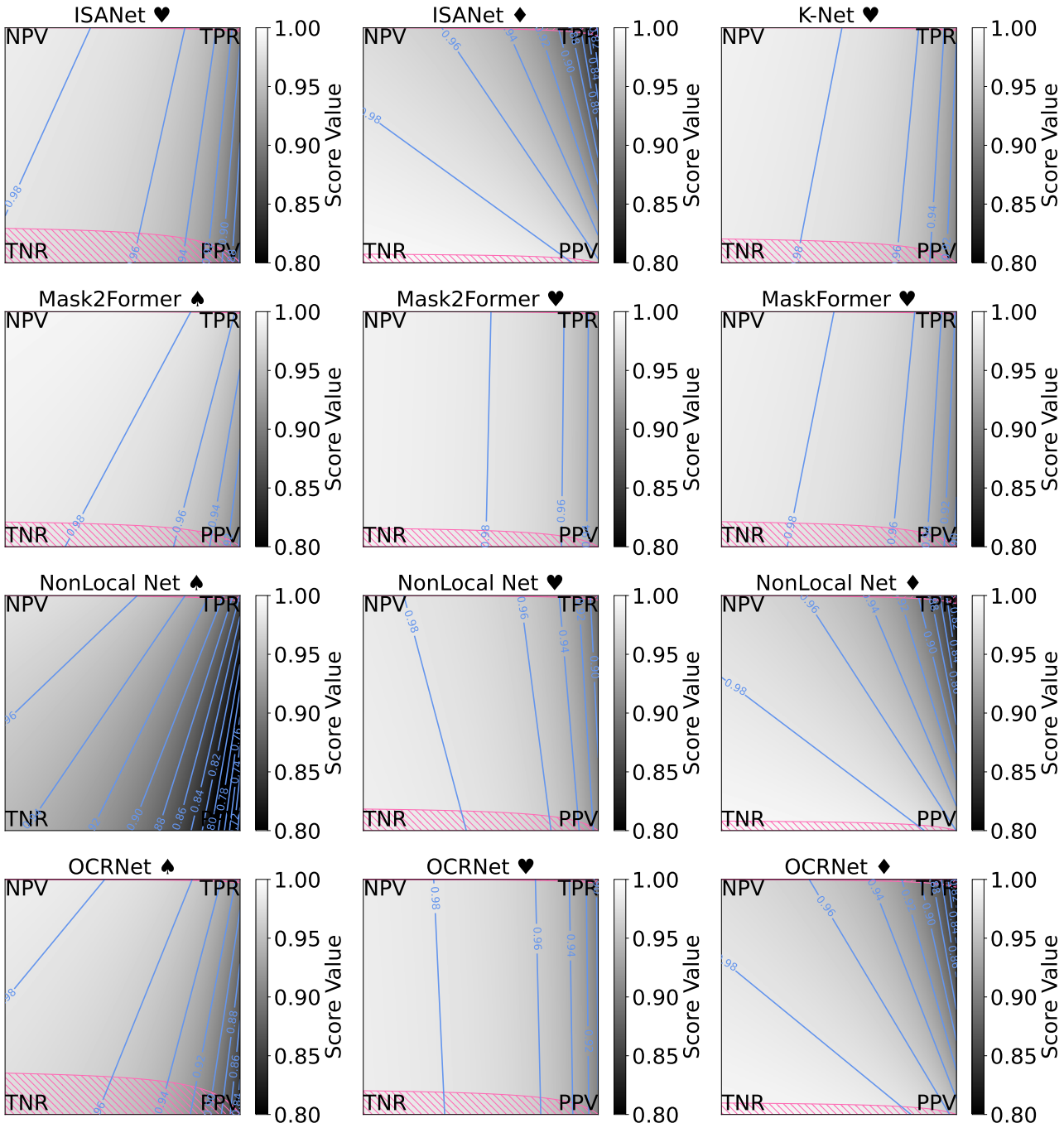
Value Tile: Using the tile to show the canonical ranking scores values for each entity

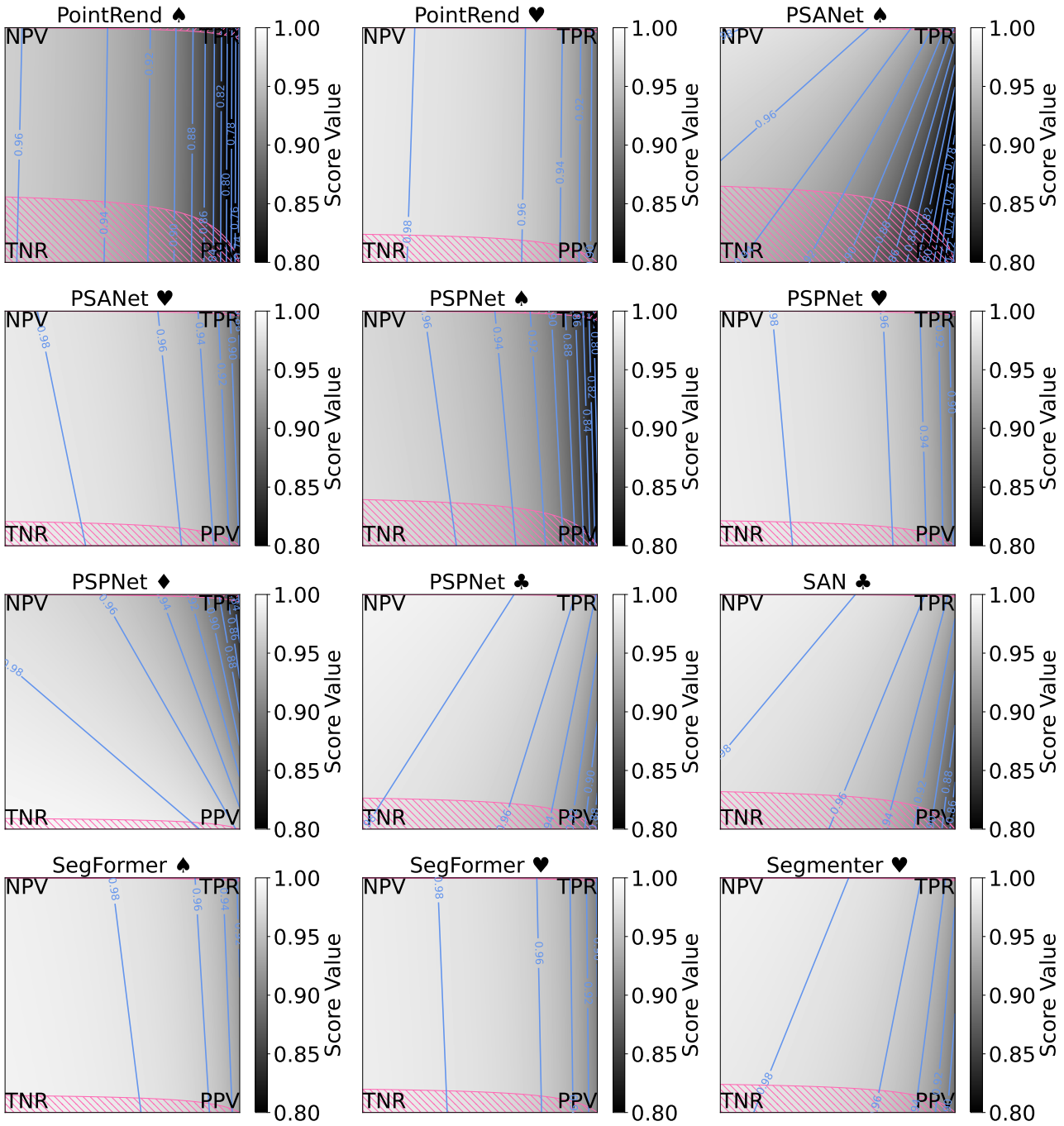


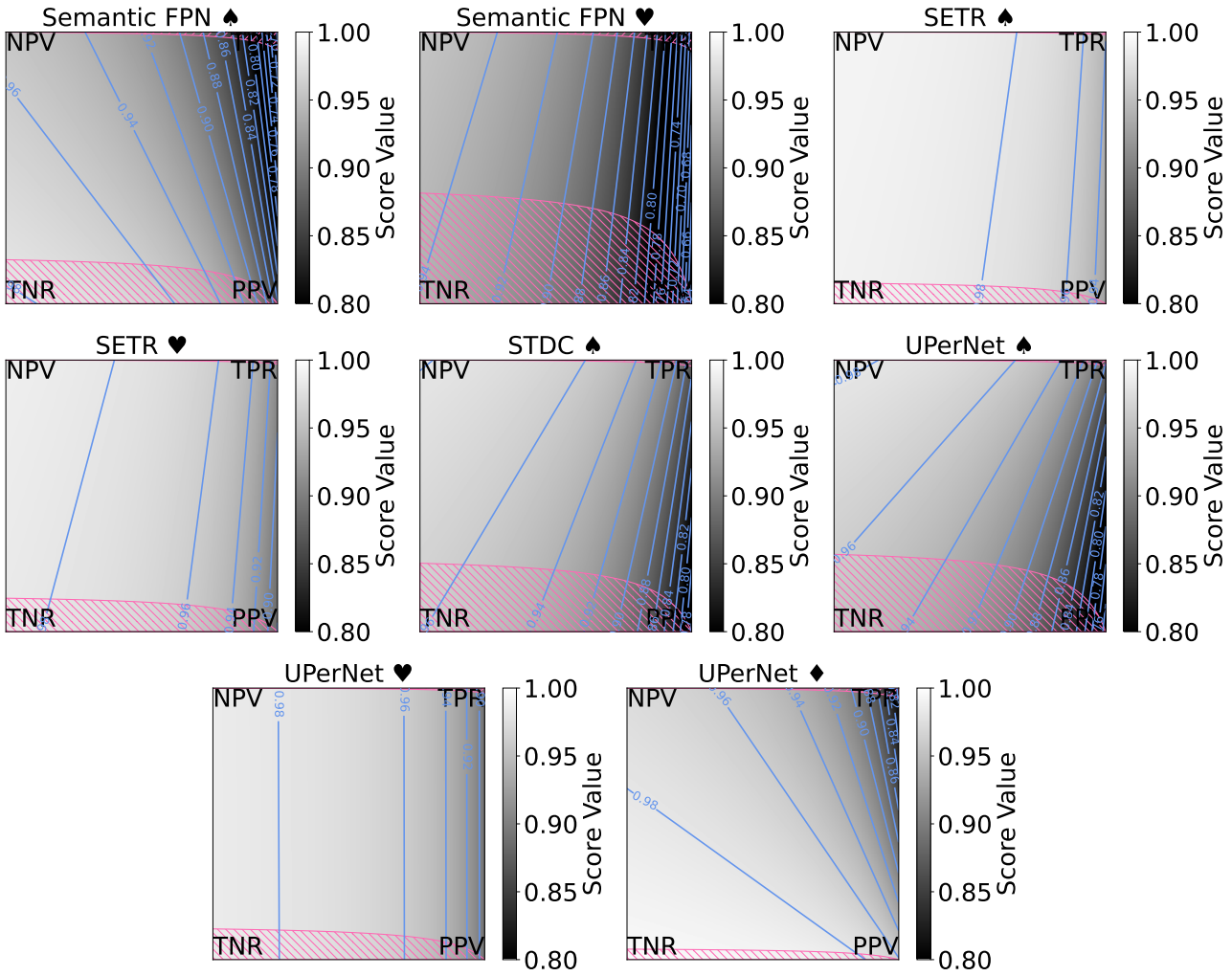

BiSeNetV1 ♣
BiSeNetV2 ♠
CCNet ♠
CCNet ♥
CCNet ♦
CGNet ♠
DANet ♠
DANet ♥
DANet ♦
DeepLabV3+ ♠
DeepLabV3+ ♥
DeepLabV3+ ♦
NPV
TPR
TNR
PPV
Score Value
1.00
0.95
0.90
0.85
0.80


DeepLabV3 ♠
DeepLabV3 ♥
DeepLabV3 ♦
DeepLabV3 ♣
DMNet ♠
DMNet ♥
DNLNet ♠
DNLNet ♥
DPT ♥
EMANet ♠
EncNet ♠
EncNet ♥
NPV
TPR
TNR
PPV
Score Value
1.00
0.95
0.90
0.85
0.80

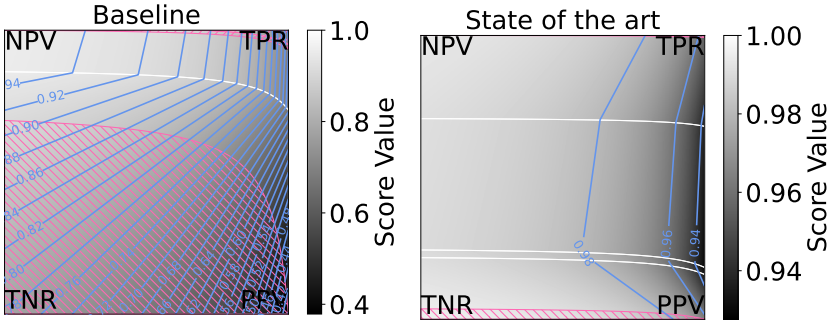

ERFNet ♠
Fast-SCNN ♠
FastFCN ♠
FastFCN ♥
FCN ♠
FCN ♥
FCN ♦
GCNet ♠
GCNet ♥
GCNet ♦
ICNet ♠
ISANet ♠
NPV
TPR
TNR
PPV
Score Value
1.00
0.95
0.90
0.85
0.80


ISANet ♥
ISANet ♦
K-Net ♥
Mask2Former ♠
Mask2Former ♥
MaskFormer ♥
NonLocal Net ♠
NonLocal Net ♥
NonLocal Net ♦
OCRNet ♠
OCRNet ♥
OCRNet ♦
NPV
TPR
TNR
PPV
Score Value
1.00
0.95
0.90
0.85
0.80

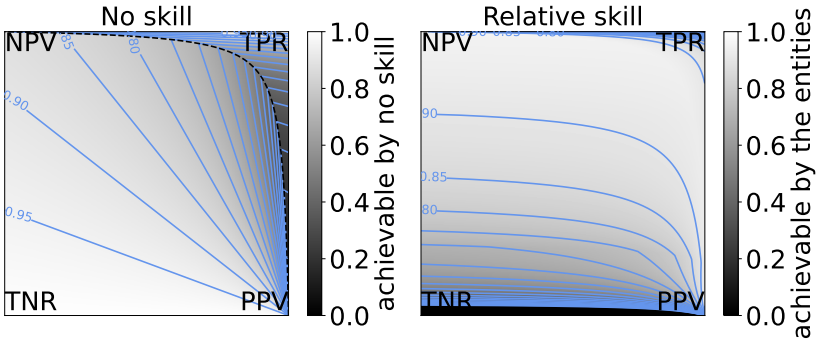

PointRend ♠
PointRend ♥
PSANet ♠
PSANet ♥
PSPNet ♠
PSPNet ♥
PSPNet ♦
PSPNet ♣
SAN ♣
SegFormer ♠
SegFormer ♥
Segmenter ♥
NPV
TPR
TNR
PPV
Score Value
1.00
0.95
0.90
0.85
0.80



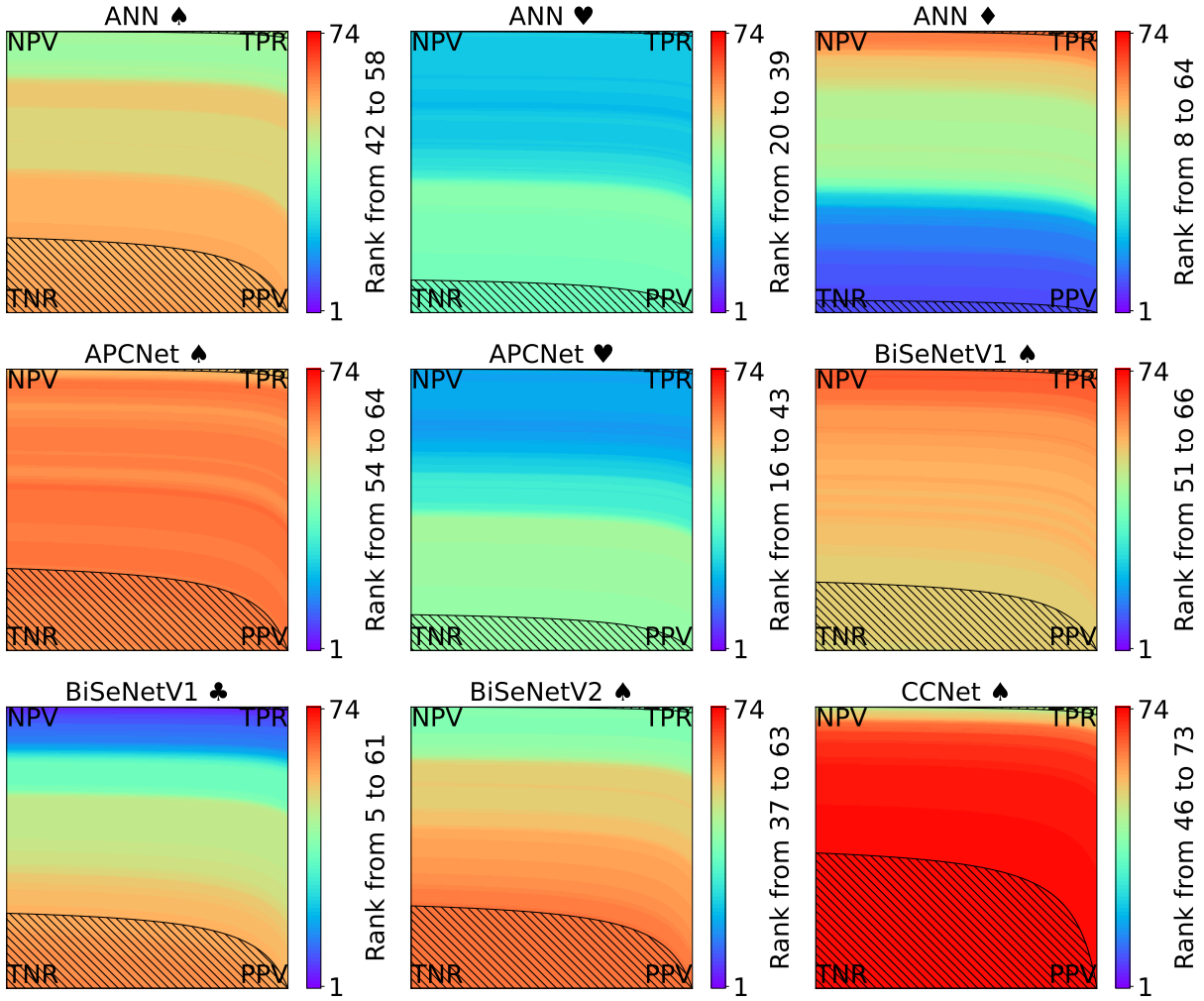
Baseline Value Tile and State-of-the-Art Value Tile: Using the tile to show the 'Baseline' and 'State of the Art'

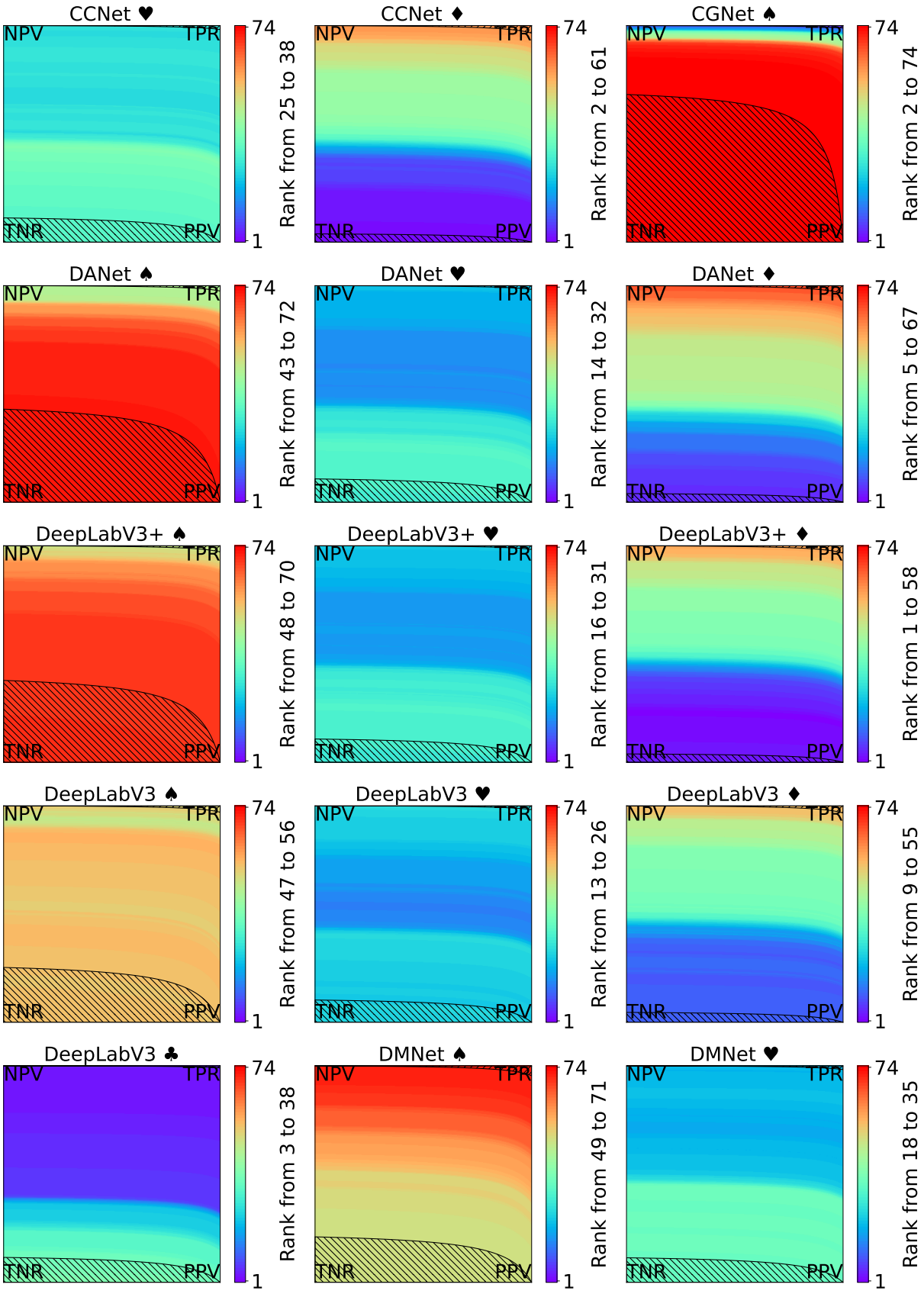


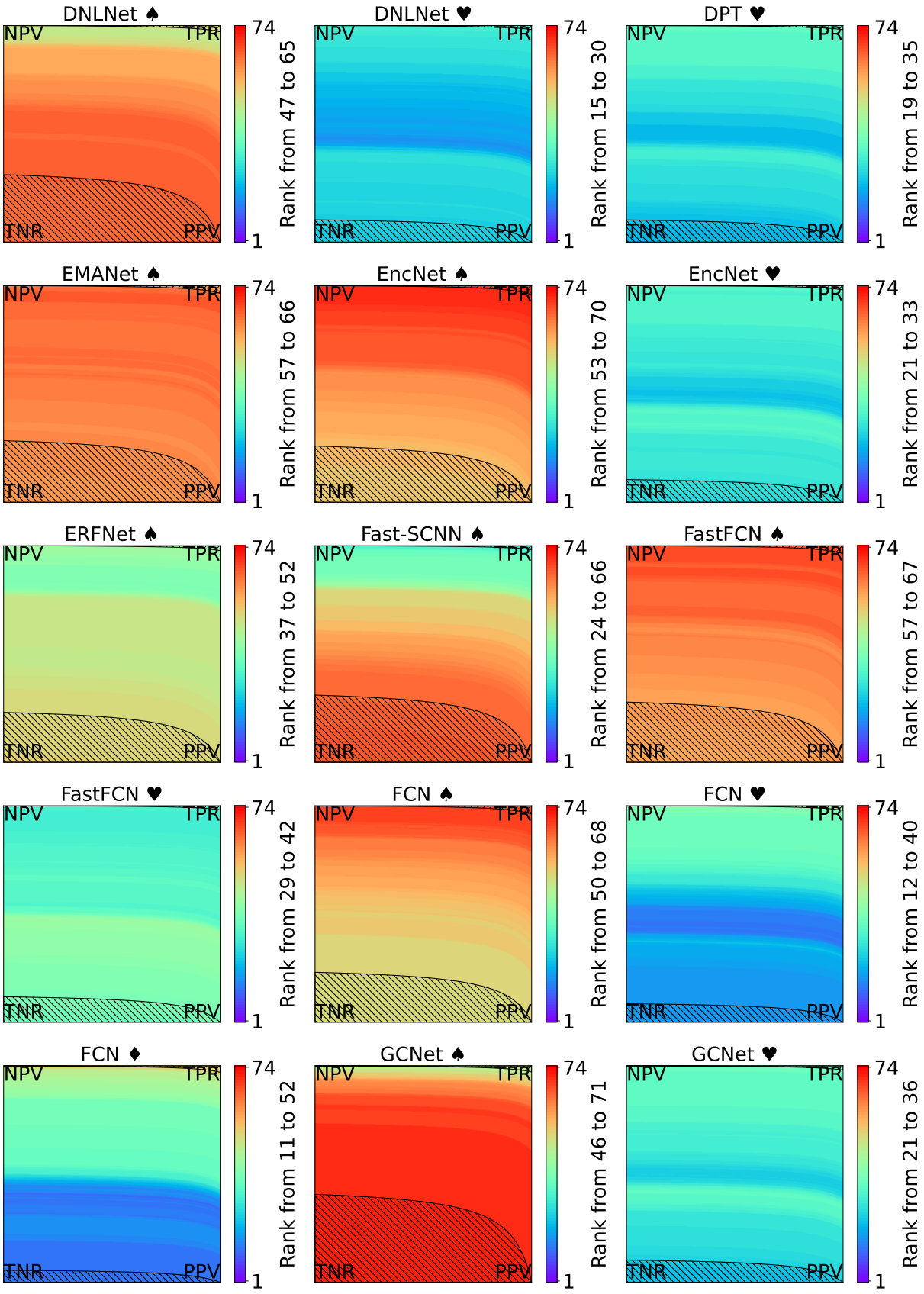
No-Skill Tile and Relative-Skill Tile: Using the tile to show the no-skill values and the relative skill values

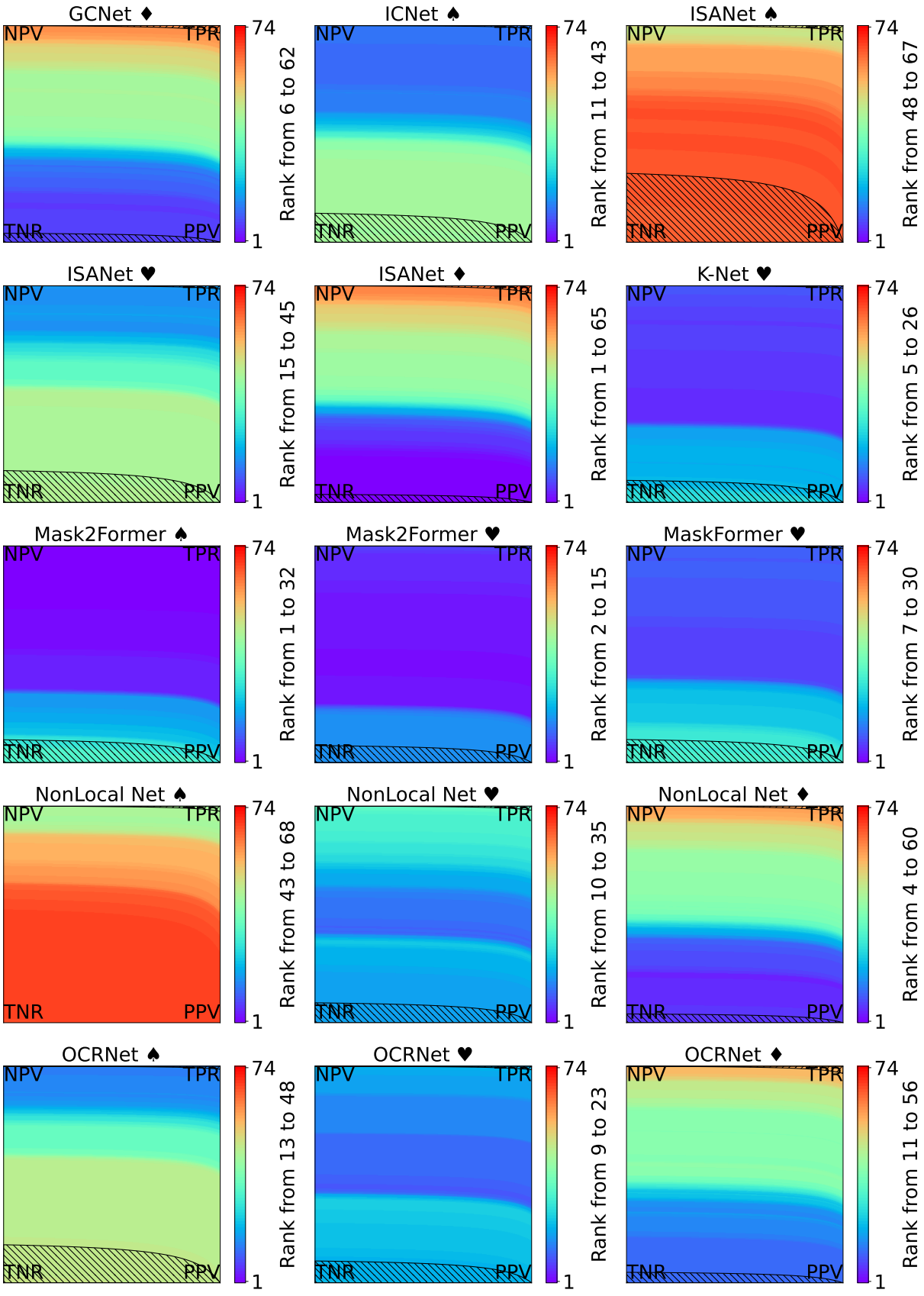


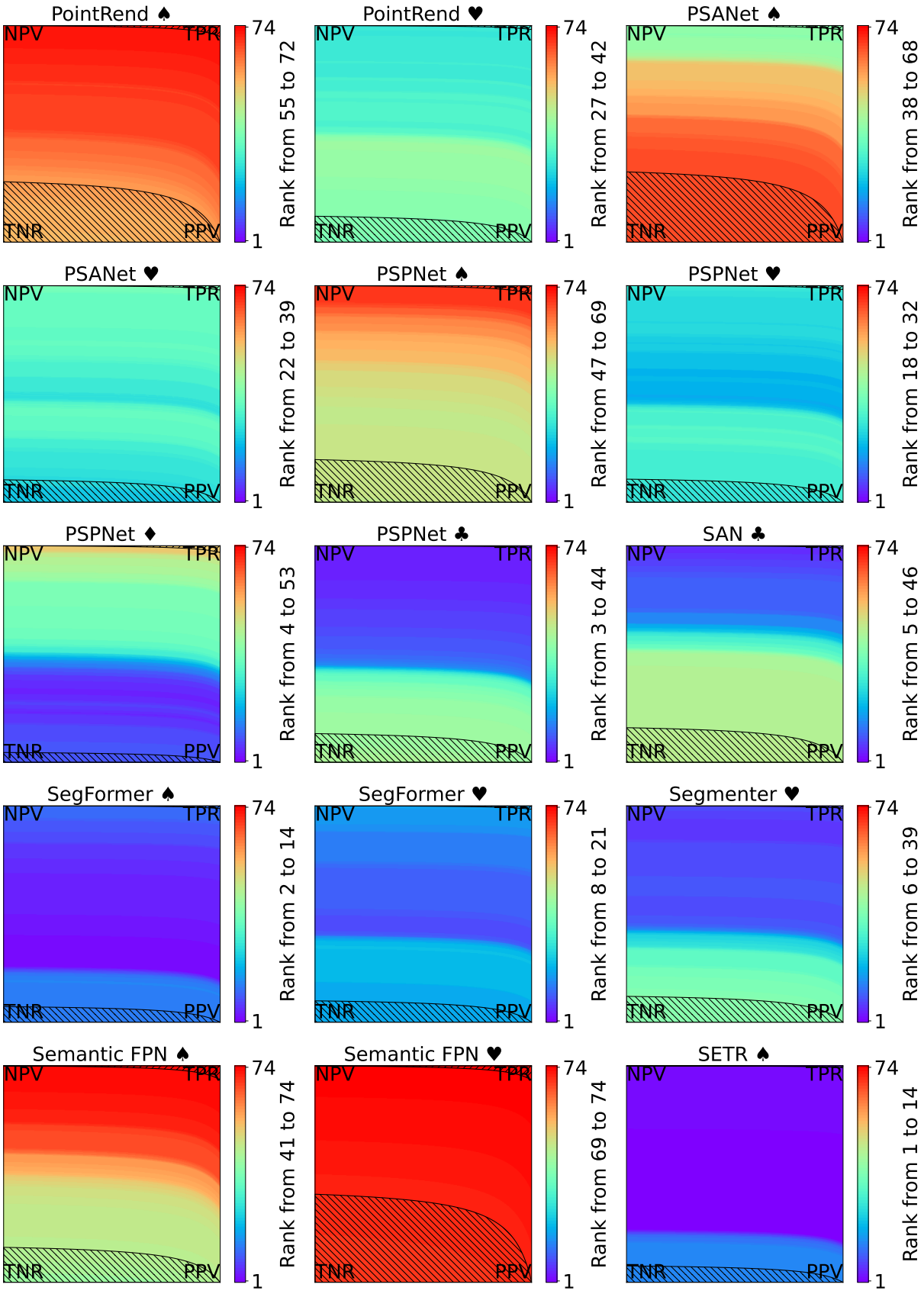
Ranking Tile: Using the tile to show the ranks for each entity

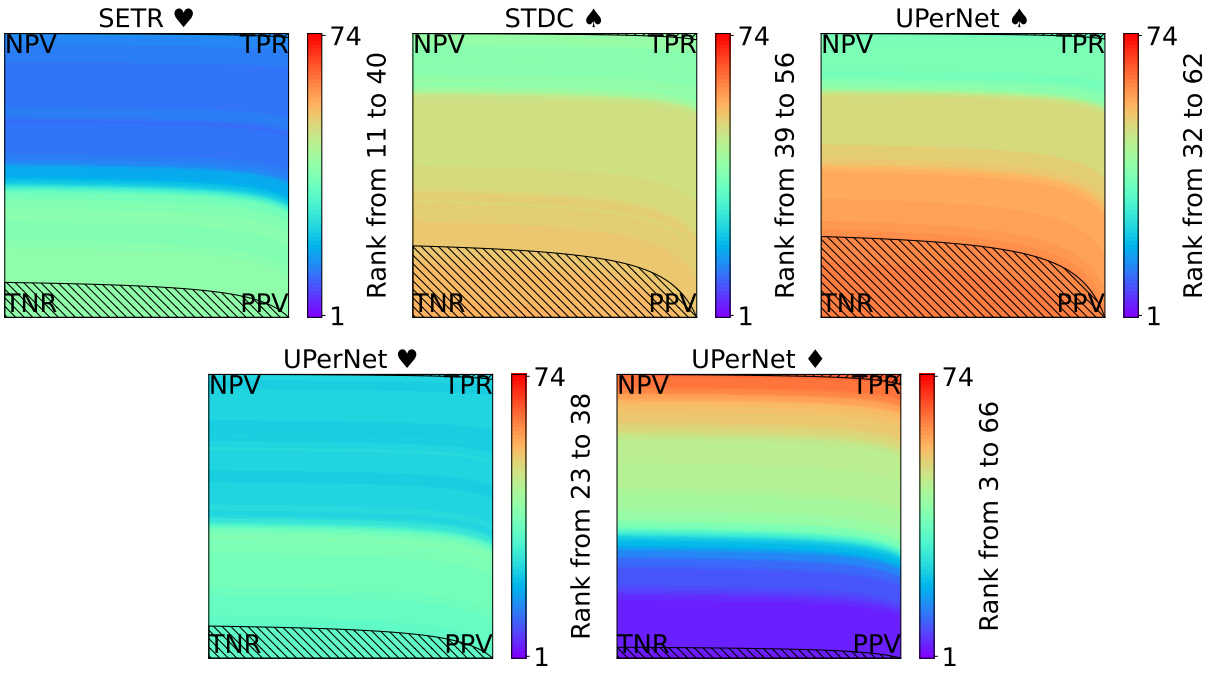



CCNet ♥
CCNet ♦
CGNet ♠
DANet ♠
DANet ♥
DANet ♦
DeepLabV3+ ♠
DeepLabV3+ ♥
DeepLabV3+ ♦
DeepLabV3 ♠
DeepLabV3 ♥
DeepLabV3 ♦
DeepLabV3 ♣
DMNet ♠
DMNet ♥
NPV
TPR
TNR
PPV
74
1
Rank from 25 to 38
Rank from 2 to 61
Rank from 2 to 74
Rank from 43 to 72
Rank from 14 to 32
Rank from 5 to 67
Rank from 48 to 70
Rank from 16 to 31
Rank from 1 to 58
Rank from 47 to 56
Rank from 13 to 26
Rank from 9 to 55
Rank from 3 to 38
Rank from 49 to 71
Rank from 18 to 35


DNLNet ♠
NPV
TPR
TNR
PPV
74
1
Rank from 47 to 65
DNLNet ♥
Rank from 15 to 30
DPT ♥
Rank from 19 to 35
EMANet ♠
Rank from 57 to 66
EncNet ♠
Rank from 53 to 70
EncNet ♥
Rank from 21 to 33
ERFNet ♠
Rank from 37 to 52
Fast-SCNN ♠
Rank from 24 to 66
FastFCN ♠
Rank from 57 to 67
FastFCN ♥
Rank from 29 to 42
FCN ♠
Rank from 50 to 68
FCN ♥
Rank from 12 to 40
FCN ♦
Rank from 11 to 52
GCNet ♠
Rank from 46 to 71
GCNet ♥
Rank from 21 to 36


GCNet ♦
NPV
TPR
TNR
PPV
74
1
Rank from 6 to 62
ICNet ♠
NPV
TPR
TNR
PPV
74
1
Rank from 11 to 43
ISANet ♠
NPV
TPR
TNR
PPV
74
1
Rank from 48 to 67
ISANet ♥
NPV
TPR
TNR
PPV
74
1
Rank from 15 to 45
ISANet ♦
NPV
TPR
TNR
PPV
74
1
Rank from 1 to 65
K-Net ♥
NPV
TPR
TNR
PPV
74
1
Rank from 5 to 26
Mask2Former ♠
NPV
TPR
TNR
PPV
74
1
Rank from 1 to 32
Mask2Former ♥
NPV
TPR
TNR
PPV
74
1
Rank from 2 to 15
MaskFormer ♥
NPV
TPR
TNR
PPV
74
1
Rank from 7 to 30
NonLocal Net ♠
NPV
TPR
TNR
PPV
74
1
Rank from 43 to 68
NonLocal Net ♥
NPV
TPR
TNR
PPV
74
1
Rank from 10 to 35
NonLocal Net ♦
NPV
TPR
TNR
PPV
74
1
Rank from 4 to 60
OCRNet ♠
NPV
TPR
TNR
PPV
74
1
Rank from 13 to 48
OCRNet ♥
NPV
TPR
TNR
PPV
74
1
Rank from 9 to 23
OCRNet ♦
NPV
TPR
TNR
PPV
74
1
Rank from 11 to 56


PointRend ♠
NPV
TPR
TNR
PPV
74
1
Rank from 55 to 72
PointRend ♥
Rank from 27 to 42
PSANet ♠
Rank from 38 to 68
PSANet ♥
Rank from 22 to 39
PSPNet ♠
Rank from 47 to 69
PSPNet ♥
Rank from 18 to 32
PSPNet ♦
Rank from 4 to 53
PSPNet ♣
Rank from 3 to 44
SAN ♣
Rank from 5 to 46
SegFormer ♠
Rank from 2 to 14
SegFormer ♥
Rank from 8 to 21
Segmenter ♥
Rank from 6 to 39
Semantic FPN ♠
Rank from 41 to 74
Semantic FPN ♥
Rank from 69 to 74
SETR ♠
Rank from 1 to 14



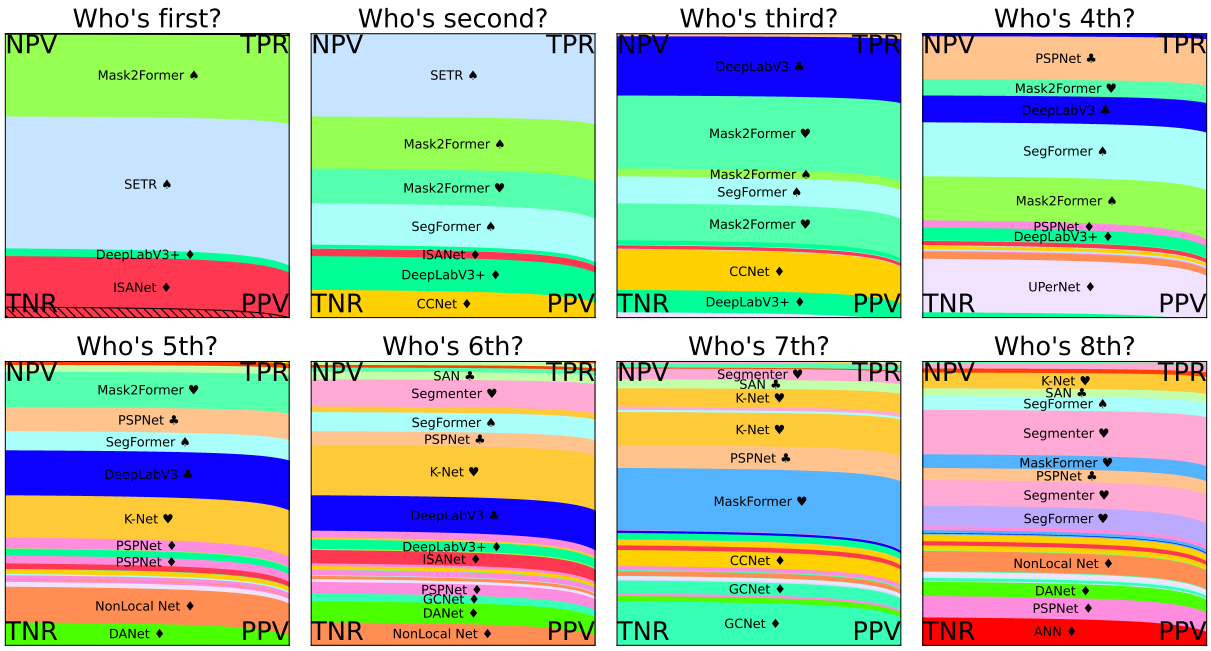
Analysis. The following entities mimimize the maximum rank:

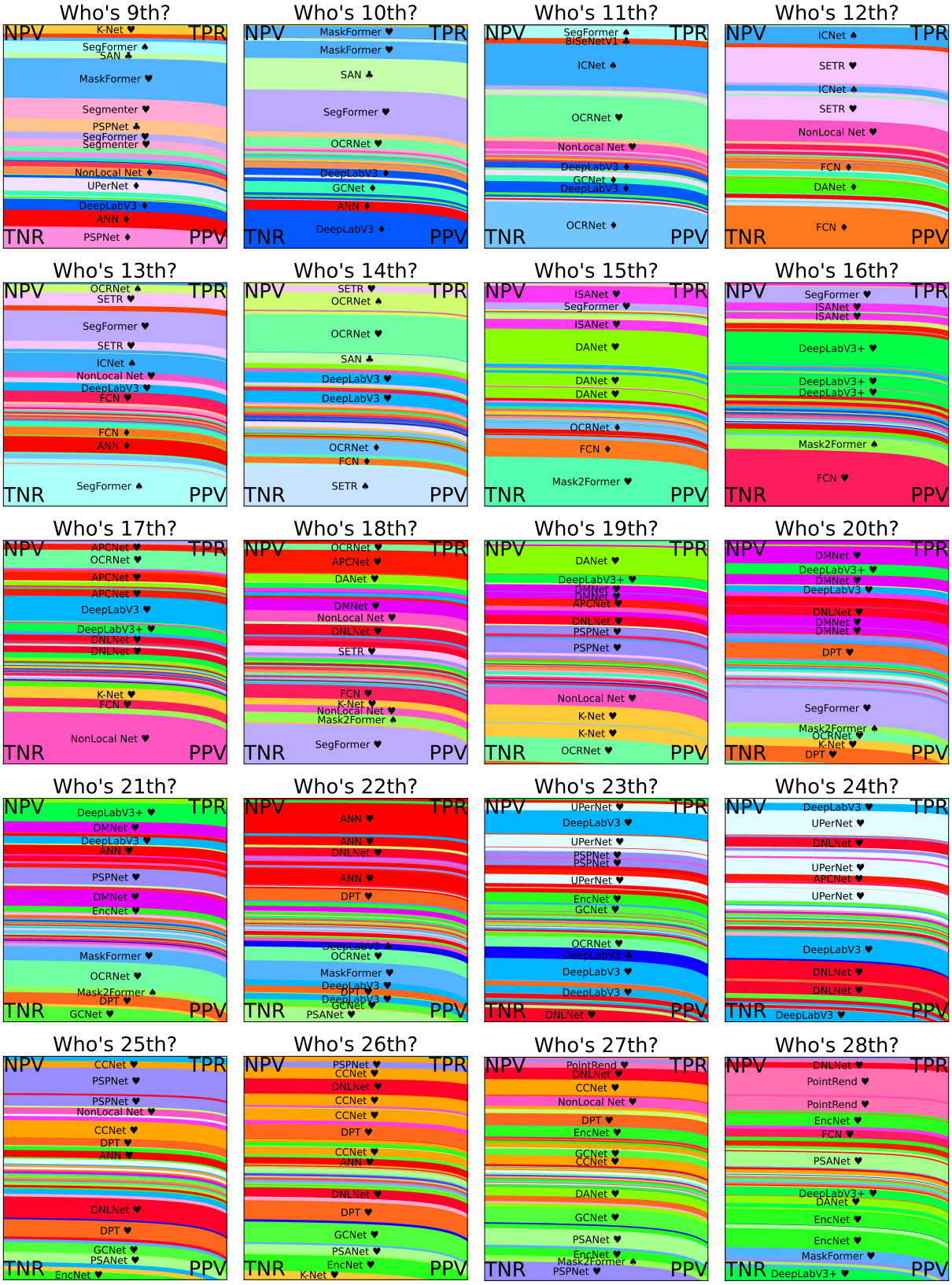
- SegFormer ♠ (max rank: 14, mean rank: 6.8558)
- SETR ♠ (max rank: 14, mean rank: 4.0495)

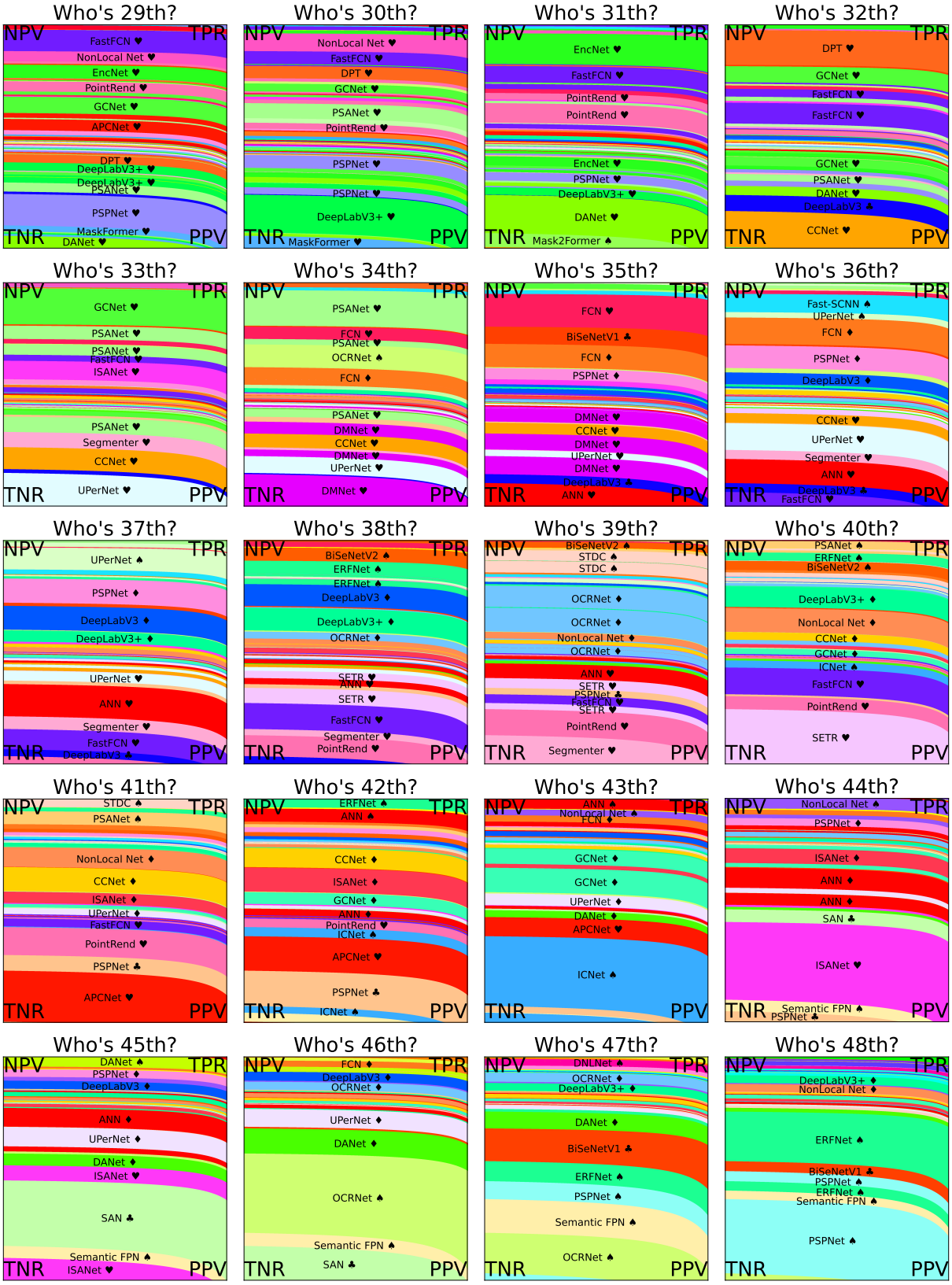
Among them, the following entities have the lowest mean rank:

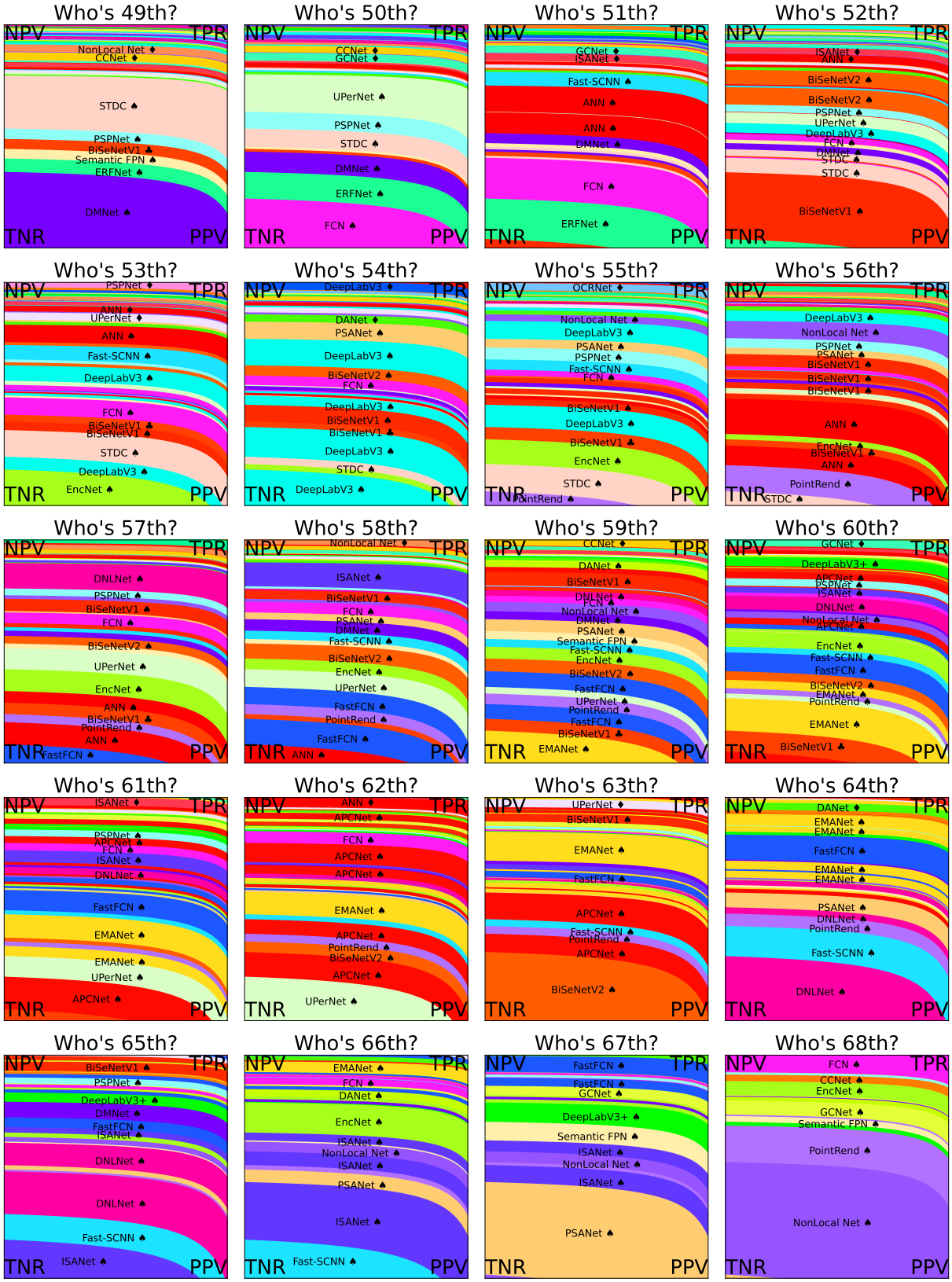
- SETR ♠

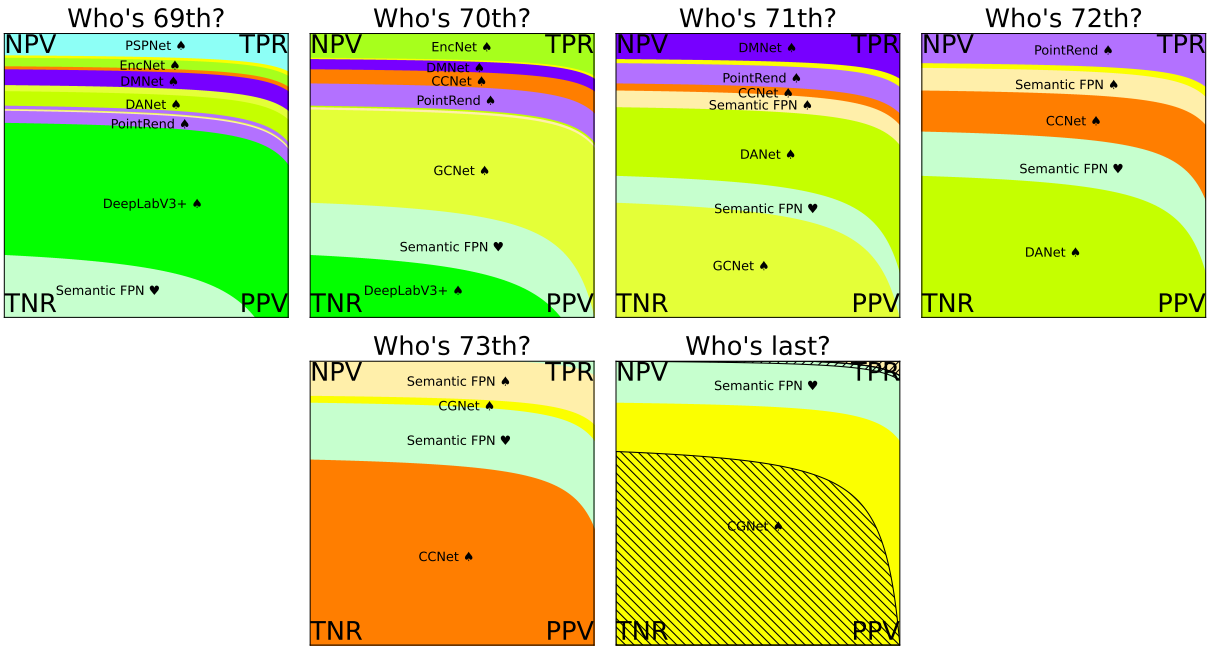
Entity Tile: Using the tile to show the entities for each rank




Who's 9th?
NPV TPR
K-Net ♥
SegFormer ♠
SAN ♣
MaskFormer ♥
Segmenter ♥
PSPNet ♣
SegFormer ♥
Segmenter ♥
NonLocal Net ♦
UPerNet ♦
DeepLabV3 ♦
ANN ♦
TNR PSPNet ♦ PPV
Who's 10th?
NPV TPR
MaskFormer ♥
MaskFormer ♥
SAN ♣
SegFormer ♥
OCRNet ♥
DeepLabV3 ♦
GCNet ♦
ANN ♦
TNR DeepLabV3 ♦ PPV
Who's 11th?
NPV TPR
SegFormer ♠
BiSeNetV1 ♣
ICNet ♠
OCRNet ♥
NonLocal Net ♥
DeepLabV3 ♦
GCNet ♦
DeepLabV3 ♦
TNR OCRNet ♦ PPV
Who's 12th?
NPV TPR
ICNet ♠
SETR ♥
ICNet ♠
SETR ♥
NonLocal Net ♥
FCN ♦
DANet ♦
TNR FCN ♦ PPV
Who's 13th?
NPV TPR
OCRNet ♠
SETR ♥
SegFormer ♥
SETR ♥
ICNet ♠
NonLocal Net ♥
DeepLabV3 ♥
FCN ♥
FCN ♦
ANN ♦
TNR SegFormer ♠ PPV
Who's 14th?
NPV TPR
SETR ♥
OCRNet ♠
OCRNet ♥
SAN ♣
DeepLabV3 ♥
DeepLabV3 ♥
OCRNet ♦
FCN ♦
TNR SETR ♠ PPV
Who's 15th?
NPV TPR
ISANet ♥
SegFormer ♥
ISANet ♥
DANet ♥
DANet ♥
DANet ♥
OCRNet ♦
FCN ♦
TNR Mask2Former ♥ PPV
Who's 16th?
NPV TPR
SegFormer ♥
ISANet ♥
ISANet ♥
DeepLabV3+ ♥
DeepLabV3+ ♥
DeepLabV3+ ♥
Mask2Former ♠
FCN ♥
TNR PPV
Who's 17th?
NPV TPR
APCNet ♥
OCRNet ♥
APCNet ♥
APCNet ♥
DeepLabV3 ♥
DeepLabV3+ ♥
DNLNet ♥
DNLNet ♥
K-Net ♥
FCN ♥
TNR NonLocal Net ♥ PPV
Who's 18th?
NPV TPR
OCRNet ♥
APCNet ♥
DANet ♥
DMNet ♥
NonLocal Net ♥
DNLNet ♥
SETR ♥
FCN ♥
K-Net ♥
NonLocal Net ♥
Mask2Former ♠
TNR SegFormer ♥ PPV
Who's 19th?
NPV TPR
DANet ♥
DeepLabV3+ ♥
DMNet ♥
DMNet ♥
APCNet ♥
DNLNet ♥
PSPNet ♥
PSPNet ♥
NonLocal Net ♥
K-Net ♥
K-Net ♥
TNR OCRNet ♥ PPV
Who's 20th?
NPV TPR
DMNet ♥
DeepLabV3+ ♥
DMNet ♥
DeepLabV3 ♥
DNLNet ♥
DMNet ♥
DMNet ♥
DPT ♥
SegFormer ♥
Mask2Former ♠
OCRNet ♥
K-Net ♥
TNR DPT ♥ PPV
Who's 21th?
NPV TPR
DeepLabV3+ ♥
DMNet ♥
DeepLabV3 ♥
ANN ♥
PSPNet ♥
DMNet ♥
EncNet ♥
MaskFormer ♥
OCRNet ♥
Mask2Former ♠
DPT ♥
TNR GCNet ♥ PPV
Who's 22th?
NPV TPR
ANN ♥
ANN ♥
DNLNet ♥
ANN ♥
DPT ♥
DeepLabV3 ♠
OCRNet ♥
MaskFormer ♥
DeepLabV3 ♥
DPT ♥
DeepLabV3 ♥
GCNet ♥
TNR PSANet ♥ PPV
Who's 23th?
NPV TPR
UPerNet ♥
DeepLabV3 ♥
UPerNet ♥
PSPNet ♥
PSPNet ♥
UPerNet ♥
EncNet ♥
GCNet ♥
OCRNet ♥
DeepLabV3 ♣
DeepLabV3 ♥
DeepLabV3 ♥
TNR DNLNet ♥ PPV
Who's 24th?
NPV TPR
DeepLabV3 ♥
UPerNet ♥
DNLNet ♥
UPerNet ♥
APCNet ♥
UPerNet ♥
DeepLabV3 ♥
DNLNet ♥
DNLNet ♥
TNR DeepLabV3 ♥ PPV
Who's 25th?
NPV TPR
CCNet ♥
PSPNet ♥
PSPNet ♥
NonLocal Net ♥
CCNet ♥
DPT ♥
ANN ♥
DNLNet ♥
DPT ♥
GCNet ♥
PSANet ♥
TNR EncNet ♥ PPV
Who's 26th?
NPV TPR
PSPNet ♥
CCNet ♥
DNLNet ♥
CCNet ♥
CCNet ♥
DPT ♥
CCNet ♥
ANN ♥
DNLNet ♥
DPT ♥
GCNet ♥
PSANet ♥
EncNet ♥
TNR K-Net ♥ PPV
Who's 27th?
NPV TPR
PointRend ♥
DNLNet ♥
CCNet ♥
NonLocal Net ♥
DPT ♥
EncNet ♥
GCNet ♥
CCNet ♥
DANet ♥
GCNet ♥
PSANet ♥
EncNet ♥
Mask2Former ♠
TNR PSPNet ♥ PPV
Who's 28th?
NPV TPR
DNLNet ♥
PointRend ♥
PointRend ♥
EncNet ♥
FCN ♥
PSANet ♥
DeepLabV3+ ♥
DANet ♥
EncNet ♥
EncNet ♥
MaskFormer ♥
TNR DeepLabV3+ ♥ PPV


Who's 29th?
NPV TPR
FastFCN ♥
NonLocal Net ♥
EncNet ♥
PointRend ♥
GCNet ♥
APCNet ♥
DPT ♥
DeepLabV3+ ♥
DeepLabV3+ ♥
PSANet ♥
PSPNet ♥
MaskFormer ♥
DANet ♥
TNR PPV
Who's 30th?
NPV TPR
NonLocal Net ♥
FastFCN ♥
DPT ♥
GCNet ♥
PSANet ♥
PointRend ♥
PSPNet ♥
PSPNet ♥
DeepLabV3+ ♥
MaskFormer ♥
TNR PPV
Who's 31th?
NPV TPR
EncNet ♥
FastFCN ♥
PointRend ♥
PointRend ♥
EncNet ♥
PSPNet ♥
DeepLabV3+ ♥
DANet ♥
Mask2Former ♠
TNR PPV
Who's 32th?
NPV TPR
DPT ♥
GCNet ♥
FastFCN ♥
FastFCN ♥
GCNet ♥
PSANet ♥
DANet ♥
DeepLabV3 ♣
CCNet ♥
TNR PPV
Who's 33th?
NPV TPR
GCNet ♥
PSANet ♥
PSANet ♥
FastFCN ♥
ISANet ♥
PSANet ♥
Segmenter ♥
CCNet ♥
UPerNet ♥
TNR PPV
Who's 34th?
NPV TPR
PSANet ♥
FCN ♥
PSANet ♥
OCRNet ♠
FCN ♦
PSANet ♥
DMNet ♥
CCNet ♥
DMNet ♥
UPerNet ♥
DMNet ♥
TNR PPV
Who's 35th?
NPV TPR
FCN ♥
BiSeNetV1 ♣
FCN ♦
PSPNet ♦
DMNet ♥
CCNet ♥
DMNet ♥
UPerNet ♥
DMNet ♥
DeepLabV3 ♣
ANN ♥
TNR PPV
Who's 36th?
NPV TPR
Fast-SCNN ♠
UPerNet ♠
FCN ♦
PSPNet ♦
DeepLabV3 ♦
CCNet ♥
UPerNet ♥
Segmenter ♥
ANN ♥
DeepLabV3 ♣
FastFCN ♥
TNR PPV
Who's 37th?
NPV TPR
UPerNet ♠
PSPNet ♦
DeepLabV3 ♦
DeepLabV3+ ♦
UPerNet ♥
ANN ♥
Segmenter ♥
FastFCN ♥
DeepLabV3 ♣
TNR PPV
Who's 38th?
NPV TPR
BiSeNetV2 ♠
ERFNet ♠
ERFNet ♠
DeepLabV3 ♦
DeepLabV3+ ♦
OCRNet ♦
SETR ♥
ANN ♥
SETR ♥
FastFCN ♥
Segmenter ♥
PointRend ♥
TNR PPV
Who's 39th?
NPV TPR
BiSeNetV2 ♠
STDC ♠
STDC ♠
OCRNet ♦
OCRNet ♦
NonLocal Net ♦
OCRNet ♦
ANN ♥
SETR ♥
PSPNet ♣
FastFCN ♥
SETR ♥
PointRend ♥
Segmenter ♥
TNR PPV
Who's 40th?
NPV TPR
PSANet ♠
ERFNet ♠
BiSeNetV2 ♠
DeepLabV3+ ♦
NonLocal Net ♦
CCNet ♦
GCNet ♦
ICNet ♠
FastFCN ♥
PointRend ♥
SETR ♥
TNR PPV
Who's 41th?
NPV TPR
STDC ♠
PSANet ♠
NonLocal Net ♦
CCNet ♦
ISANet ♦
UPerNet ♦
FastFCN ♥
PointRend ♥
PSPNet ♣
APCNet ♥
TNR PPV
Who's 42th?
NPV TPR
ERFNet ♠
ANN ♣
CCNet ♦
ISANet ♦
GCNet ♦
ANN ♦
PointRend ♥
ICNet ♠
APCNet ♥
PSPNet ♣
ICNet ♠
TNR PPV
Who's 43th?
NPV TPR
ANN ♠
NonLocal Net ♠
FCN ♦
GCNet ♦
GCNet ♦
UPerNet ♦
DANet ♦
APCNet ♥
ICNet ♠
TNR PPV
Who's 44th?
NPV TPR
NonLocal Net ♠
PSPNet ♦
ISANet ♦
ANN ♦
ANN ♦
SAN ♣
ISANet ♥
Semantic FPN ♠
PSPNet ♣
TNR PPV
Who's 45th?
NPV TPR
DANet ♠
PSPNet ♦
DeepLabV3 ♦
ANN ♦
UPerNet ♦
DANet ♦
ISANet ♥
SAN ♣
Semantic FPN ♠
ISANet ♥
TNR PPV
Who's 46th?
NPV TPR
FCN ♦
DeepLabV3 ♦
OCRNet ♦
UPerNet ♦
DANet ♦
OCRNet ♠
Semantic FPN ♠
SAN ♣
TNR PPV
Who's 47th?
NPV TPR
DNLNet ♠
OCRNet ♦
DeepLabV3+ ♦
DANet ♦
BiSeNetV1 ♣
ERFNet ♠
PSPNet ♠
Semantic FPN ♠
OCRNet ♠
TNR PPV
Who's 48th?
NPV TPR
DeepLabV3+ ♦
NonLocal Net ♦
ERFNet ♠
BiSeNetV1 ♣
PSPNet ♠
ERFNet ♠
Semantic FPN ♠
PSPNet ♠
TNR PPV

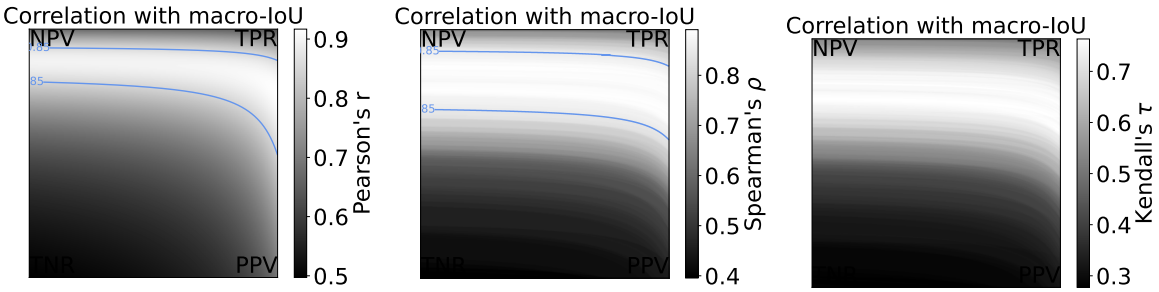

Who's 49th?
NPV TPR
NonLocal Net ♦
CCNet ♦
STDC ♠
PSPNet ♠
BiSeNetV1 ♣
Semantic FPN ♠
ERFNet ♠
DMNet ♠
TNR PPV
Who's 50th?
NPV TPR
CCNet ♦
GCNet ♦
UPerNet ♠
PSPNet ♠
STDC ♠
DMNet ♠
ERFNet ♠
FCN ♠
TNR PPV
Who's 51th?
NPV TPR
GCNet ♦
ISANet ♦
Fast-SCNN ♠
ANN ♠
ANN ♠
DMNet ♠
FCN ♠
ERFNet ♠
TNR PPV
Who's 52th?
NPV TPR
ISANet ♦
ANN ♦
BiSeNetV2 ♠
BiSeNetV2 ♠
PSPNet ♠
UPerNet ♠
DeepLabV3 ♠
FCN ♠
DMNet ♠
STDC ♠
STDC ♠
BiSeNetV1 ♠
TNR PPV
Who's 53th?
NPV TPR
PSPNet ♦
ANN ♠
UPerNet ♦
ANN ♠
Fast-SCNN ♠
DeepLabV3 ♠
FCN ♠
BiSeNetV1 ♣
BiSeNetV1 ♠
STDC ♠
DeepLabV3 ♠
EncNet ♠
TNR PPV
Who's 54th?
NPV TPR
DeepLabV3 ♦
DANet ♦
PSANet ♠
DeepLabV3 ♠
BiSeNetV2 ♠
FCN ♠
DeepLabV3 ♠
BiSeNetV1 ♠
BiSeNetV1 ♣
DeepLabV3 ♠
STDC ♠
DeepLabV3 ♠
TNR PPV
Who's 55th?
NPV TPR
OCRNet ♦
NonLocal Net ♠
DeepLabV3 ♠
PSANet ♠
PSPNet ♠
Fast-SCNN ♠
FCN ♠
BiSeNetV1 ♠
DeepLabV3 ♠
BiSeNetV1 ♣
EncNet ♠
STDC ♠
PointRend ♠
TNR PPV
Who's 56th?
NPV TPR
DeepLabV3 ♠
NonLocal Net ♠
PSPNet ♠
PSANet ♠
BiSeNetV1 ♠
BiSeNetV1 ♠
BiSeNetV1 ♠
ANN ♠
EncNet ♠
BiSeNetV1 ♣
ANN ♠
PointRend ♠
STDC ♠
TNR PPV
Who's 57th?
NPV TPR
DNLNet ♠
PSPNet ♠
BiSeNetV1 ♠
FCN ♠
BiSeNetV2 ♠
UPerNet ♠
EncNet ♠
ANN ♠
BiSeNetV1 ♣
PointRend ♠
ANN ♠
FastFCN ♠
TNR PPV
Who's 58th?
NPV TPR
NonLocal Net ♦
ISANet ♦
BiSeNetV1 ♠
FCN ♠
PSANet ♠
DMNet ♠
Fast-SCNN ♠
BiSeNetV2 ♠
EncNet ♠
UPerNet ♠
FastFCN ♠
PointRend ♠
FastFCN ♠
ANN ♠
TNR PPV
Who's 59th?
NPV TPR
CCNet ♦
DANet ♠
BiSeNetV1 ♠
DNLNet ♠
FCN ♠
NonLocal Net ♠
DMNet ♠
PSANet ♠
Semantic FPN ♠
Fast-SCNN ♠
EncNet ♠
BiSeNetV2 ♠
FastFCN ♠
UPerNet ♠
PointRend ♠
FastFCN ♠
BiSeNetV1 ♣
EMANet ♠
TNR PPV
Who's 60th?
NPV TPR
GCNet ♦
DeepLabV3+ ♠
APCNet ♠
PSPNet ♠
ISANet ♠
DNLNet ♠
NonLocal Net ♠
APCNet ♠
EncNet ♠
Fast-SCNN ♠
FastFCN ♠
BiSeNetV2 ♠
EMANet ♠
PointRend ♠
EMANet ♠
BiSeNetV1 ♣
TNR PPV
Who's 61th?
NPV TPR
ISANet ♦
PSPNet ♠
APCNet ♠
FCN ♠
ISANet ♠
DNLNet ♠
FastFCN ♠
EMANet ♠
EMANet ♠
UPerNet ♠
APCNet ♠
TNR PPV
Who's 62th?
NPV TPR
ANN ♦
APCNet ♠
FCN ♠
APCNet ♠
APCNet ♠
EMANet ♠
APCNet ♠
PointRend ♠
BiSeNetV2 ♣
APCNet ♠
UPerNet ♠
TNR PPV
Who's 63th?
NPV TPR
UPerNet ♦
BiSeNetV1 ♠
EMANet ♠
FastFCN ♠
APCNet ♠
Fast-SCNN ♠
PointRend ♠
APCNet ♠
BiSeNetV2 ♠
TNR PPV
Who's 64th?
NPV TPR
DANet ♦
EMANet ♠
EMANet ♠
FastFCN ♠
EMANet ♠
EMANet ♠
PSANet ♠
DNLNet ♠
PointRend ♠
Fast-SCNN ♠
DNLNet ♠
TNR PPV
Who's 65th?
NPV TPR
BiSeNetV1 ♠
PSPNet ♠
DeepLabV3+ ♠
DMNet ♠
FastFCN ♠
ISANet ♠
DNLNet ♠
DNLNet ♠
Fast-SCNN ♠
ISANet ♠
TNR PPV
Who's 66th?
NPV TPR
EMANet ♠
FCN ♠
DANet ♠
EncNet ♠
ISANet ♠
NonLocal Net ♠
ISANet ♠
PSANet ♠
ISANet ♠
Fast-SCNN ♠
TNR PPV
Who's 67th?
NPV TPR
FastFCN ♠
FastFCN ♠
GCNet ♠
DeepLabV3+ ♠
Semantic FPN ♠
ISANet ♠
NonLocal Net ♠
ISANet ♠
PSANet ♠
TNR PPV
Who's 68th?
NPV TPR
FCN ♠
CCNet ♠
EncNet ♠
GCNet ♠
Semantic FPN ♠
PointRend ♠
NonLocal Net ♠
TNR PPV



Analysis. The following entities are ranked first:

- DeepLabV3+ ♦ (2.66% of the tile)
- ISANet ♦ (20.52% of the tile)
- Mask2Former ♠ (29.85% of the tile)
- SETR ♠ (46.97% of the tile)

Correlation Tile: Using the tile to show the rank and linear correlations with the mean-IoU



Analysis for the linear correlation with Pearson's r.

- In 93.9% of the zone where $r \geq 0.85$ in the tile, the best is Mask2Former ♠
- In 6.1% of the zone where $r \geq 0.85$ in the tile, the best is SETR ♠

Analysis for the rank correlation with Spearman's ρ.

- In 81.5% of the zone where $\rho \geq 0.85$ in the tile, the best is Mask2Former ♠
- In 18.5% of the zone where $\rho \geq 0.85$ in the tile, the best is SETR ♠

Analysis for the rank correlation with Kendall's τ.

- WARNING: There is no zone where $\tau \geq 0.85$ in the tile! Maybe do you want to change the threshold?