

Appendix

A Experiment Setup

A.1 Datasets, DNNs, Hyperparameters

Datasets: We use five datasets in our experiments to test different black-box DNNs-under-test. First, as a proof-of-concept for our overall approach, we use the CelebA (Liu et al., 2015) dataset with a rich collection of 40 facial attributes (metadata) for 202 599 images of celebrity faces. We used the aligned PNG images provided by the authors, which have a resolution of 178×218 pixels. Next, for pedestrian detection tasks, we consider BDD100k (Yu et al., 2020), Cityscapes (Cordts et al., 2016), RailSem19 (Zendel et al., 2019), and EuroCity Persons dataset (Braun et al., 2019). In these datasets, we focus only on the pedestrian class in the 2D-bounding box and semantic segmentation tasks. For BDD100k, we consider the predefined validation set of 10k samples of resolution 1280×720 , while in Cityscapes and RailSem19, due to their smaller dataset sizes, we use the entire train and validation sets containing 3475 (the test set is not considered due to the lack of GT) and 8500 samples with image resolutions 2048×1024 and 1920×1080 respectively. In EuroCity Persons, we separately analyse the “day” and “night” subsets provided in the dataset. For each, we used combined training and validation splits, comprising 28,158 images for the day subset and 4,992 images for the night subset.

Models (DNNs-under-test): We evaluate five black-box models for ODD aligned systematic weaknesses. For the first experiment, we consider the publicly available ViT-B-16 (Dosovitskiy et al., 2021) model pre-trained on ImageNet21k (Ridnik et al., 2021) from the python library timm.⁸ Second, we use the pre-trained publicly available Faster R-CNN (Ren et al., 2015) object detector with ConvNeXt-T (Liu et al., 2022) backbone. The model weights are available on the BDD100k model Zoo.⁹ Third, for the Cityscapes dataset, we use a pre-trained SETR PUP (Zheng et al., 2021) semantic segmentation model. The model weights are available on the mmsegmentation codebase.¹⁰ From the railway domain, we use a PanopticFCN (Li et al., 2021) model, which has been trained by an industrial partner on a large proprietary dataset also including RailSem19 (Zendel et al., 2019). We consider this as a complete black box and have no details on the concrete training procedure. We additionally evaluate the YOLOv11m (Jocher & Qiu, 2024) model to illustrate the performance of our approach on larger models. For the autonomous datasets, the black-box model performance per-object (i.e., pedestrian) is measured by the intersection-over-union (IoU). For our experiments, we consider an IoU greater than 0 as a true positive and the rest as false negatives to simplify the evaluation.

Parameters of CLIP and SliceLine: For metadata generation, we use a pre-trained CLIP (Radford et al., 2021) with image encoder (ViT-L/14 (Dosovitskiy et al., 2021)). For SliceLine, we use a python implementation and choose default α and σ values of 0.95 and $n/100$ where n defines the size of the structured data as proposed in Sagadeeva & Boehm (2021). For the synthetic data experiment, we incrementally increase k from 1 to 60.

A.2 Synthetic Data Generation Parameters

The purpose of the synthetic data experiment is to evaluate the algorithm with control over the quality of labeling and without the influence of correlations. Therefore, we build a tabular dataset with 9 “real” semantic dimensions (dim1, ..., dim9) containing 200,000 rows. For each of these dimensions, we generate a synthetic dimension as a proxy for labeling by CLIP. All dimensions contain binary attributes. For the first five dimensions, the distribution of true attributes is imbalanced, i.e., only 5% of overall samples ([8000, 9000, 10000, 11000, 12000]), respectively. The other dimensions are balanced between both attributes. The final column contains errors simulating the **DuT** performance. Next, we define a set of slices and induce errors for each of the slices. For our experiments, we induce the following errors: {dim1: 0.19, dim2 & dim3: 0.18, dim3: 0.23, dim4: 0.3, dim5: 0.07, dim6: 0.04, dim7: 0.01, dim8: 0.05, dim9: 0.02}. As we

⁸<https://github.com/huggingface/pytorch-image-models>

⁹<https://github.com/SysCV/bdd100k-models/tree/main/det>

¹⁰<https://github.com/open-mmlab/mmdetection>

have 100 runs for different labeling qualities, we introduce random fluctuations between -0.01 and 0.01 to these error values. The choice of errors and number of dimensions is to align the synthetic data with the CelebA experiment and also to effectively induce errors. If all dimensions contribute roughly equally to the error rate, no strong signal for a specific slice, in contrast to the others, could be found. We generate 100 runs each for 3 different labeling qualities. That is, we generate the “observed” metadata from the “real” one using a random predefined “precision” value. For good quality, this precision value to detect attribute 1 of each dimension is sampled from a uniform distribution between 0.8 and 1.0. For attribute 0, we sample between 0.8 and 1.0. For medium quality and attribute 1, we sample from 0.4 and 0.6 and for attribute 0 between 0.4 and 0.7. For bad quality, for attribute 1, we sample between 0.1 and 0.4 and for attribute 0 between 0.3 and 0.6.

A.3 Human-understandable Dimensions

Herrmann et al. (2022) have proposed ontologies for different dynamic objects (e.g., pedestrians) to build ODDs for AD vehicles. Although these proposed ontologies do not yet completely capture all safety-relevant features, they provide a reference to the direction safety experts intend to take to build evidences for safety augmentations of AD vehicles. To enable such a formulation of evidence, we performed our experiments on a subset of the concepts discussed in these ontologies as shown in tables 5 and 6. In the case of BDD100k, as information about occlusion is provided in the dataset, we combine our generated metadata with this additional information. For the CelebA experiment, as the input distribution is not directly related to the AD domain, we consider semantic concepts that are more suitable for this dataset as shown in table 5. Similar to Gannamaneni et al. (2023), we encode the input image using the CLIP image encoder. For CelebA dataset, we encode the entire input image, while for the AD experiments, we encode individual pedestrian crops as a single input. We consider each semantic dimension and its corresponding attributes to generate metadata for an input image.

Semantic dimension	Attributes	
Gender	Male	Female
Pale-skin	True	False
Age	Young	Adult
Beard	True	False
Goatee	True	False
Bald	True	False
Wearing-Hat	True	False
Wearing-Eyeglasses	True	False
Smiling	True	False

Table 5: The ODD used for the CelebA experiment. The first column represents the different semantic dimensions (in analogy to safety-relevant features). For each dimension, different attributes are considered and generated as metadata using our metadata generation process.

A.4 SliceLine Workflow

SliceLine works on individual errors e_i of data samples i . These, in the original work, can be defined as $e_i = 1 - p_i$ with the DuT predicted probability p_i for the correct class. In the remainder, we make the simplifying assumption that $e_i \in \{0, 1\}$ indicates whether i was classified correctly, $e_i = 0$, or not, $e_i = 1$. The workflow of SliceLine to identify weak slices is as follows: Initially, for depth level 1, a breadth search is performed on all attributes in the metadata such that only single features form a slice (e.g., a slice containing all data points with condition ($gender : male$)). Checks are performed over these slices to ensure that thresholds are met (e.g., minimum slice size specified via some parameter σ). Next, based on the slice scores from eq. (5), the slices are ordered, and a list of top-k weak slices is populated. The hyperparameter α in eq. (5) allows us to weight the size of the slice as well as the error signal. At depth level 2 and above, combinations of two attributes are chosen to form a slice (e.g., slice containing all data points with condition ($gender = male$)&($occlusion = (0.9, 1.0]$)). The list of weak slices is updated after each depth level. The

Semantic dimension	Attributes	
Gender	Male	Female
Skin color	White	Dark
Age	Young	Adult
Clothing color	Bright-color	Dark-color
Blurry	True	False
Occlusion [†]	True	False
Construction-worker [‡]	True	False
Size	10 quantile binned values of bounding box pixel area	

Table 6: A sample ontology for pedestrians used in our AD dataset experiments. The first column represents the different semantic dimensions (safety relevant features). For each dimension, different attributes are considered and generated as metadata using our metadata generation process. Metadata that is generated from CLIP but from available through other sources (e.g., GT) is not considered noisy and, therefore, we do not perform precision and recall estimation by human sampling. [†] Occlusion is available as GT from the BDD100k and EuroCity Persons dataset and we only consider it in the corresponding experiments. [‡] RailSem19 dataset contains several images where construction-workers are present near railway tracks. Therefore, we additionally consider this dimension for CLIP labeling to identify if models have weaknesses identifying construction workers. Size of pedestrian is estimated by calculating product of bounding width and height.

maximum depth level is a hyperparameter. In addition, pruning steps are also performed at each depth level in the original implementation. The conditions for pruning have a monotonicity property, which ensures that all potential sub-slices of a pruned slice would also fulfill the pruning condition. Due to the limited sizes of the ODDs for our experiment, we do not consider the pruning step in our implementation. Once the maximum depth level has been reached, the algorithm is terminated and the final list of top- k weak slices is available.

$$\text{Scoring Function}(\mathcal{S}) = \alpha \frac{e^{|\mathcal{S} - e|_{\mathcal{D}}}}{e^{|\mathcal{D}|}} - (1 - \alpha) \frac{|\mathcal{D}| - |\mathcal{S}|}{|\mathcal{S}|} \quad (5)$$

A.5 Scalability

Here, we discuss the scalability of our algorithm w.r.t. size of the **DuT** M , the dataset \mathcal{D} , the number of metadata dimensions and their attributes \mathcal{Z} , and the maximum number of semantic combinations considered simultaneously, i.e., maximum search depth or level ℓ in SliceLine. First, considering the size of the **DuT**, an inference step is performed on each sample s to obtain the predictions and calculate the errors. As inference requires a constant time t_m per sample, the total time complexity is $\mathcal{O}(|\mathcal{D}|)$. Similarly, considering the size of the dataset \mathcal{D} , the time complexity is linear. Furthermore, after inference on **DuT**, the generation of metadata with \mathcal{G} requires another round of inference as \mathcal{G} (e.g., CLIP) performs both embedding and classification steps. Since the images from \mathcal{D} are only embedded once with \mathcal{G} and metadata is generated w.r.t. dimensions defined in \mathcal{Z} , for a fixed \mathcal{Z} , the time complexity for this is also linear $\mathcal{O}(|\mathcal{D}|)$ assuming t_g is the constant time taken to generate metadata per sample. Next, regarding scalability of the metadata \mathcal{Z} , we consider two factors: the number of dimensions in \mathcal{Z} and the number of attributes per dimension, i.e., the cardinality. For a fixed dataset, for each sample s , metadata is generated with \mathcal{G} independently per dimension. Therefore, the total time taken increases linearly as given by $\mathcal{O}(|\mathcal{Z}|)$.

Turning to the slice discovery problem, given the performance and generated metadata, we generally expect a non-linear scaling with the number $|\mathcal{Z}|$ of considered semantic dimensions. For simplicity, we assume that each of the dimensions is binary. However, since we limit the depth (denoted as “level”), i.e., the number of combinations that are simultaneously considered, the number of slices is not exponential but algebraic in

$|\mathcal{Z}|$. This can be seen as follows: when considering all slices at a fixed level, the number of combinations is

$$N(|\mathcal{Z}|, \text{level}) = 2^{\text{level}} \binom{|\mathcal{Z}|}{\text{level}} \quad (6)$$

as we take all combinations of “level” number of dimensions from \mathcal{Z} and within these combinations have 2^{level} ways of distributing the absence or presence of the attribute. If we consider the total number of slices up to a given level ℓ , the following upper bound holds:

$$\sum_{k=0}^{\ell} N(|\mathcal{Z}|, k) = \sum_{k=0}^{\ell} 2^k \binom{|\mathcal{Z}|}{k} \leq \sum_{k=0}^{\ell} 2^k \frac{|\mathcal{Z}|^k}{k!} \leq \sum_{k=0}^{\ell} (2|\mathcal{Z}|)^k = \frac{(2|\mathcal{Z}|)^{\ell+1} - 1}{2|\mathcal{Z}| - 1} \sim (2|\mathcal{Z}|)^{\ell}, \quad (7)$$

where we used that

$$\binom{a}{b} = \frac{a!}{b!(a-b)!} = \frac{1}{b!} \prod_{k=a-b+1}^a k \leq \frac{1}{b!} a^b \leq a^b. \quad (8)$$

This shows that for a fixed level, the runtime of SliceLine, with complexity $\mathcal{O}((2|\mathcal{Z}|)^{\text{level}})$ grows polynomially with the number of semantic dimensions. Despite this scaling, each individual calculation in SliceLine is fast, implying that, in practice, the largest amount of absolute computation is incurred during the metadata generation and **DuT** inference.

B Derivations

B.1 Derivation of $p(e|\mathcal{C})$ and $p(e|\mathcal{S})$

To derive eq. (1) and eq. (2) from section 3, we first consider the joined probability $p(e, \mathcal{C}, \mathcal{S})$, where e denotes the DuT error, \mathcal{C} labeling, and \mathcal{S} the ground truth for some semantic attribute. Using Bayes’ Theorem we can rewrite this as

$$p(e, \mathcal{C}, \mathcal{S}) = p(e|\mathcal{C}, \mathcal{S})p(\mathcal{C}, \mathcal{S}) = p(e|\mathcal{C}, \mathcal{S})p(\mathcal{C}|\mathcal{S})p(\mathcal{S}). \quad (9)$$

Looking additionally at marginal distributions

$$p(e, \mathcal{S}) = \sum_{\mathcal{C}} p(e, \mathcal{C}, \mathcal{S}) = p(\mathcal{S}) \sum_{\mathcal{C}} p(e|\mathcal{C}, \mathcal{S})p(\mathcal{C}|\mathcal{S}), \quad (10)$$

where the sum goes over all possible values \mathcal{C} can take. We can write the conditional error probability (or rate if considered over finite data) as

$$p(e|\mathcal{S}) = \sum_{\mathcal{C}} p(e|\mathcal{C}, \mathcal{S})p(\mathcal{C}|\mathcal{S}) \quad (11)$$

At this point, using that \mathcal{C} takes only binary values, which, for brevity, we denote as \mathcal{C} if the attribute was detected and as $\neg\mathcal{C}$ else,¹¹ we can expand the sum:

$$p(e|\mathcal{S}) = p(e|\mathcal{C}, \mathcal{S})p(\mathcal{C}|\mathcal{S}) + p(e|\neg\mathcal{C}, \mathcal{S})p(\neg\mathcal{C}|\mathcal{S}) \quad (12)$$

Within this expression, we can identify the recall

$$r_{\mathcal{C}} \equiv p(\mathcal{C}|\mathcal{S}) \quad (13)$$

of the labeling method, that is the probability we will obtain correct identification of the semantic attribute given its presence. Using further the normalisation property

$$1 = \sum_{\mathcal{C}} p(\mathcal{C}|\mathcal{S}) \rightarrow p(\neg\mathcal{C}|\mathcal{S}) = 1 - p(\mathcal{C}|\mathcal{S}), \quad (14)$$

¹¹This is a slight over-use of the notation, but it is apparent from context whether \mathcal{C} is meant as the random variable for the labeling, or as its value in the sense of positive detection.

we arrive at the originally presented eq. (1):

$$p(e|\mathcal{S}) = r_c p(e|\mathcal{C}, \mathcal{S}) + (1 - r_c) p(e|\neg\mathcal{C}, \mathcal{S}) \quad (15)$$

Along the same lines eq. (2),

$$p(e|\mathcal{C}) \equiv p_c p(e|\mathcal{C}, \mathcal{S}) + (1 - p_c) p(e|\mathcal{C}, \neg\mathcal{S}), \quad (16)$$

can be derived, however with the identification

$$p_c = p(\mathcal{S}|\mathcal{C}), \quad (17)$$

i.e., the precision of the labeling process.

B.2 Derivation of Correction Equation

As discussed in section 3, the annotation process may not be a perfect process. Furthermore, there is no guarantee that the failure modes within this process do not overlap the failures of **DuT**, i.e., there is a possibility that some amount of correlation could exist between the errors of the annotation process and the errors of **DuT**. Therefore, we frame this using the following

$$\delta p(e|\mathcal{S}) = p(e|\neg\mathcal{C}, \mathcal{S}) - p(e|\mathcal{C}, \mathcal{S}) \quad (18)$$

By considering earlier equations and their complementary forms for $\neg\mathcal{S}$ and reducing the equation set, we obtain

$$A = \begin{pmatrix} p_c & 1 - p_c \\ 1 - p_{-c} & p_{-c} \end{pmatrix}, \quad B = \begin{pmatrix} p(e|\mathcal{C}) + (p_c) \delta p(e|\neg\mathcal{S}) \\ p(e|\neg\mathcal{C}) + (p_{-c}) \delta p(e|\mathcal{S}) \end{pmatrix},$$

$$A \begin{pmatrix} p(e|\mathcal{C}, \mathcal{S}) \\ p(e|\neg\mathcal{C}, \neg\mathcal{S}) \end{pmatrix} = B$$

Here, the $\det(A)$ is given by $p_c + p_{-c} - 1$ and the inverse of A is given by

$$A^{-1} = \frac{1}{p_c + p_{-c} - 1} \begin{pmatrix} p_{-c} & -(1 - p_c) \\ -(1 - p_{-c}) & p_c \end{pmatrix}. \quad (19)$$

Solving for the intermediate value of $p(e|\neg\mathcal{C}, \mathcal{S})$ and plugging this in eq. (15) along with eq. (18), we obtain the final equation eq. (4):

$$p(e|\mathcal{S}) = \underbrace{\frac{p(e|\mathcal{C}) p_{-c} + p(e|\neg\mathcal{C}) (p_c - 1)}{p_c + p_{-c} - 1}}_{\text{independence assumption}} + \underbrace{\delta p(e|\mathcal{S}) \left(\frac{p_c p_{-c}}{p_c + p_{-c} - 1} - r_c \right) + \delta p(e|\neg\mathcal{S}) \frac{(p_c - 1) p_{-c}}{p_c + p_{-c} - 1}}_{\text{correction terms}}. \quad (20)$$

Regarding the denominator $p_c + p_{-c} - 1$, it can be zero (or approximately zero) for some combinations of precision of the metadata annotation process. In these cases, no statement can be made on \mathcal{S} as the performance of the annotation classification does not allow separation of \mathcal{S} from the rest of the data and any observable error differences on \mathcal{C} potentially stems only from the correction factors. Besides this technical breakdown of the hypothesis, it should be pointed out that the scaling factors $\kappa_{\mathcal{S}, \neg\mathcal{S}}$ depend only on the performance of the annotation process and thus can be determined without knowing the correction factors δp themselves. While the latter are challenging to determine in practice they are rarely non-zero, even in cases where the hypothesis holds, due to fluctuations (e.g. when errors are determined on finite sample sizes). Knowing the magnitude of κ therefore allows us a degree of certainty on the statements of the hypothesis.

B.3 Quantitative Evaluation of Metadata Generation Process

Our metadata generation is a form of data labeling process. Within this work, we chose CLIP (Radford et al., 2021) to generate the metadata but know that for certain attributes of the ODDs the performance might be far below human capabilities, compare, e.g., Gannamaneni et al. (2023). To estimate the performance of our metadata generation process without large-scale evaluation or manual labeling, we take a simplifying view. For each slice \mathcal{C} containing a semantic concept identified by CLIP, for instance, images containing gender “female”, we randomly draw a few samples to create a smaller subset \mathcal{R} . Let q denote the probability that images within \mathcal{C} contain the correct semantic concept. By manually evaluating the smaller sample of images $\mathcal{R} \subset \mathcal{C}$ (drawn with replacement), we can model the posterior distribution for q using Bayes theorem, that is

$$p(q|\mathcal{R}) = \frac{p(q)p(\mathcal{R}|q)}{p(\mathcal{R})} \propto p(\mathcal{R}|q). \quad (21)$$

Therein, we assumed a flat prior, i.e. $p(q) = \text{const.}$. The probability of the observed sample \mathcal{R} is given by

$$p(\mathcal{R}|q) = \binom{n}{l} q^l (1-q)^{n-l}, \quad (22)$$

where $n = |\mathcal{R}|$ is the size of the observed sample taken from \mathcal{S} and $l \leq n$ is the number of observed positive, i.e., correct instances. The true value for q for the entire slice would describe the precision of the labeling of the concept as it is the ratio of true instances to the overall number of samples. We can approximate it using the small set using

$$p_{\text{precision}}(q|\mathcal{R}) = \frac{(n+1)!}{l!(n-l)!} q^l (1-q)^{n-l}, \quad (23)$$

where the factorials serve as the normalization. Using eq. (23), we can, therefore, determine both the expected value of q as well as our uncertainty of its value, which we report in terms of the standard deviation σ . As a side note, for values of q near 0 or 1, the Binomial distribution is asymmetric and the standard deviation is not always a faithful measure of “true” deviation. However, we compared with a quantile-based approach, taking the range from the 1/6th to 5/6th quantile, and found only minor discrepancies.

Besides estimating the precision, we are also interested in estimating the recall of the labeling process. This latter quantity is harder to evaluate as it depends both on the number of true positives and false negatives. Let P and N denote the total number of data points that are classified as containing, or respectively, as not containing, the semantic concept. Then the probability over the total number of true positives is given by $p_{\text{precision}}(q|\mathcal{R}_P)$, where \mathcal{R}_P is a random sample taken from the set \mathcal{C}_P of positively classified elements. A similar statement holds for the number of false negatives, where a sample \mathcal{R}_N from the non-detected set can be used. However, in this case, we either have to count (for l) the number of prediction errors or use the inverse outcome $1 - q$. Given that both samples are free of intersection, that is $\mathcal{R}_P \cap \mathcal{R}_N = \emptyset$, we make the assumption that the obtained probabilities q_P and q_N are independent from one another. In this case, we can formulate the recall as

$$\begin{aligned} p_{\text{recall}}(q|\mathcal{R}_P, \mathcal{R}_N) &= \int_0^1 dq_P \int_0^1 dq_N \\ &\times \delta \left(q - \frac{Pq_P}{Pq_P + N(1 - q_N)} \right) \\ &\times p_{\text{precision}}(q_P|\mathcal{R}_P) p_{\text{precision}}(q_N|\mathcal{R}_N), \end{aligned} \quad (24)$$

where we interpret $p_{\text{precision}}$ such that in both cases correct predictions are counted while δ denotes a Dirac-Delta Distribution. We evaluate this function numerically and use the results of p_{recall} in the same way as for the precision above regarding, e.g., the reported standard deviation.

B.4 Precision Sampling at different levels

In appendix B.3, we provide the framework for how precision and recall can be estimated by sampling data in slices. In this section, we present the concrete steps taken at level 1 to operationalize it and also the steps

taken to calculate precision and recall at higher levels. At level 1, in synthetic and CelebA experiments, as GT labels are available in addition to classification function \mathcal{G} labels, human evaluation of slices is not necessary. For each slice in the data, before running SliceLine, we sample with replacement ($n=60$), and using GT slice labels, calculate precision and recall based on appendix B.3. This gives us mean and standard deviations of precision and recall that can be used with eq. (4). For AD datasets, as GT labels are not available, we performed human evaluation by first taking 60 samples for each level 1 slice. The results of this are shown in table 3.

At level 2 and higher, human sampling of precisions gets very labour-intensive even if considering only 9 semantic dimensions with binary attributes. Therefore, we incorporate the parent-level precisions calculated earlier to estimate corrected errors by accounting for their contributions. From level 2 onward, we construct a composite inverse matrix,

$$A_{\mathcal{S}_1\mathcal{S}_2}^{-1} = A_{\mathcal{S}_1}^{-1} \otimes A_{\mathcal{S}_2}^{-1}, \quad (25)$$

which is a direct product of the inverse matrices given in eq. (19) for the respective semantic dimensions \mathcal{S}_1 and \mathcal{S}_2 . The direct product implies an element-wise multiplication of the differing elements of $A_{\mathcal{S}_i}^{-1}$ in all possible combinations. This approach can be understood by first considering that in the approximation the precision values in $A_{\mathcal{S}_1\mathcal{S}_2}$ are given by products of the respective precisions for $\mathcal{S}_{1,2}$ or its negations $\neg\mathcal{S}_{1,2}$. That is, for two 2×2 matrices $A_{\mathcal{S}_i}$ the resulting $A_{\mathcal{S}_1\mathcal{S}_2}$ will be 4×4 dimensional. Second, the inverse of this direct product matrix is given by the direct product of its constituent matrices, leading to eq. (25).

We can also extend the binary case of eq. (19) to a multi-class setting by taking into account that the matrices A are based on normalized confusion matrices. That is, row-wise the entries in A give the rate or probability with which the classifier \mathcal{G} will mistake a given element for an element of a foreign class. This notion easily generalizes to arbitrary classes by taking the full confusion matrix for all classes, thereby introducing all combinations besides “False Positives” or “False Negatives” from the binary case. For n classes this would result in a $n \times n$ matrix for A , the inverse of which can be used to obtain the hypothesis part of eq. (20).

C Results

C.1 Further results: CelebA Evaluations

In the synthetic data experiment, we provide the spread of errors and the precision and recall of slice recovery in comparison to an Oracle for different values of k . With the GT metadata in the CelebA dataset, we build a similar Oracle for comparison and provide similar error spread and precision and recall values of SWD-1,2,3 in fig. 5. Here, the plot depicting the spread of errors is restricted to level 1 errors for better visualization. However, the precision and recall plot is based on the full level 2 slices.

C.2 Evaluation of Top-5 Weak Slices

In this section, we provide both the quantitative and qualitative results of our experiments. For the CelebA dataset, figs. 6 to 9 contain the identified top-5 weak slices in the experiments SWD-3, DOMINO, Spotlight, and SVM FD respectively. We provide 8 samples from each of the top-5 slices found by the methods and 8 samples from the remaining data, except SVM FD which only provides 1 weak slice. In addition, we provide four slice descriptions given by DOMINO for each slice and the single slice description of SVM FD. While the actionability of our proposed approach is inherent as the identified weak slices are based on semantic concepts from the ODD, the textual descriptions from DOMINO are comparatively less useful. Furthermore, by focusing only on the samples from DOMINO, it is still hard to identify which semantic concepts uniquely constitute a slice. For example, if we consider an image from the remaining data (rightmost column), it is not straightforward to say if this image does or does not belong to any of the weak slices. Although the fifth slice does appear to capture a coherent slice, images of sports persons, the observed error $p(e|\mathcal{C})$ is significantly lower than what is identified by our approach. It is important to note that both DOMINO and SliceLine judge performance in terms of class probabilities, not false negative counts. Therefore, weak slices can have slightly better performance in terms of $p(e|\mathcal{C})$ compared to overall data, as observed for slice 3 found by DOMINO.

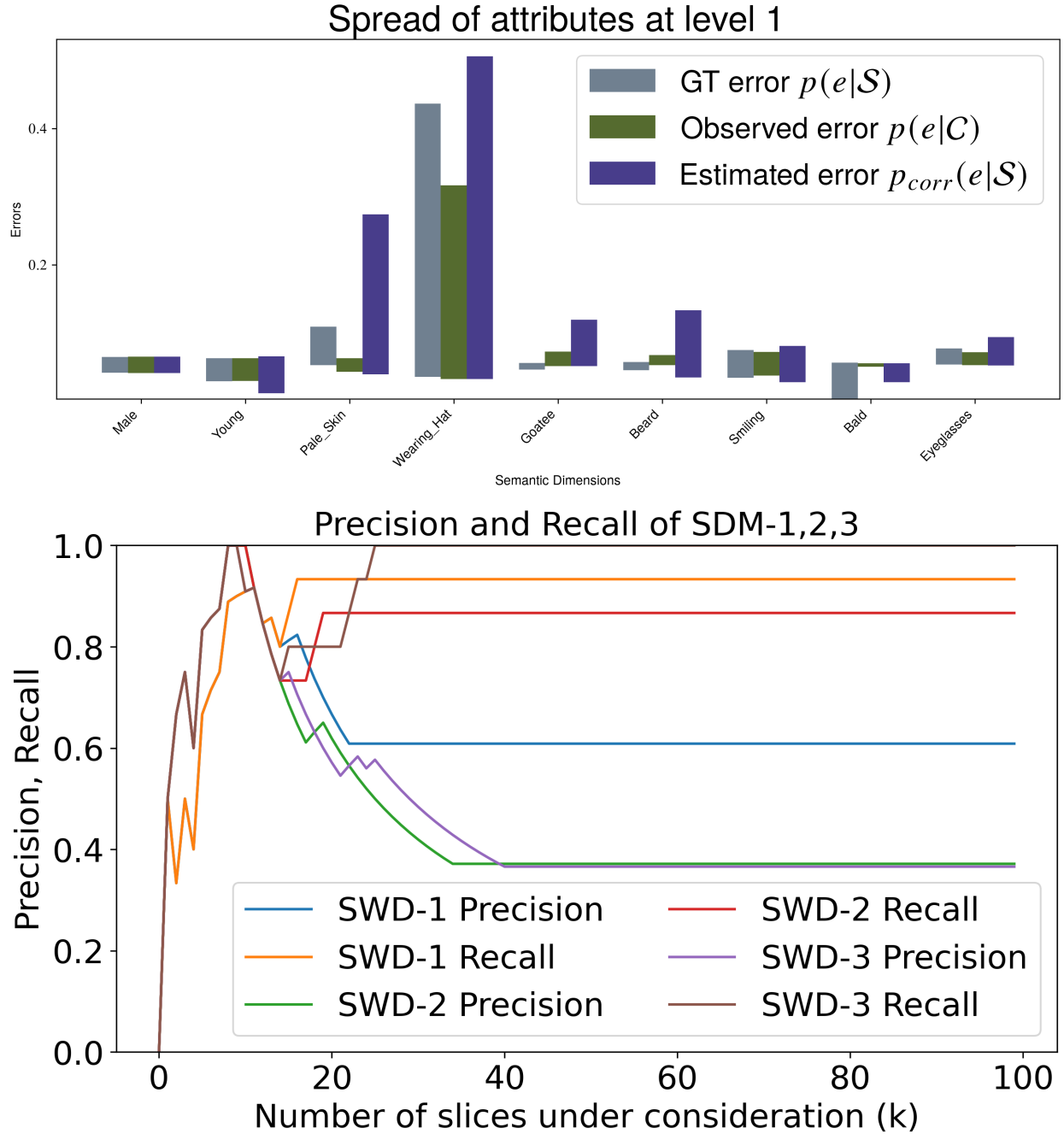


Figure 5: Similar to synthetic data experiment, we provide spread of error (top) and Precision and Recall at different levels of k for SWD-1,2,3 of algorithm 1 in comparison to the Oracle (bottom). The **DuT** is a ViT-B-16 classification model trained on ImageNet21k and evaluated on CelebA dataset. Note that here precision and recall are quality metrics of weak slice discovery and not of labeling quality.

In figs. 10 to 12, pedestrian crop samples from the top-5 weak slices obtained using our method are provided for each autonomous driving experiment. The quantitative evaluation of the top-5 slices for the three experiments can be found in tables 7 to 9.

Slice No.	$ \mathcal{S} $	$p_{\text{corr}}(e \mathcal{S})$	Avg. Perf. Degra.	Slice Description
\mathcal{S}_1	319	0.2206	-0.1636	blurry: false occluded: true
\mathcal{S}_2	508	0.2099	-0.1528	blurry: false cloth.-color: dark-color
\mathcal{S}_3	466	0.147	-0.0899	blurry: false age: adult
\mathcal{S}_4	773	0.1263	-0.0693	blurry: false
\mathcal{S}_5	582	0.1263	-0.0693	blurry: false gender: Male

Table 7: Quantitative analysis of the top-5 weak slices obtained using SWD-3 for the Faster R-CNN object detector trained and evaluated on BDD100k dataset.

Slice No.	$ \mathcal{S} $	$p_{\text{corr}}(e \mathcal{S})$	Avg. Perf. Degra.	Slice Description
\mathcal{S}_1	690	0.1046	-0.0897	age: adult skin-color: dark
\mathcal{S}_2	591	0.0921	-0.0773	skin-color: dark cloth.-color: dark-color
\mathcal{S}_3	349	0.0896	-0.0748	gender: female skin-color: dark
\mathcal{S}_4	766	0.0778	-0.0630	skin-color: dark blurry: false
\mathcal{S}_5	997	0.0594	-0.0446	skin-color: dark

Table 8: Quantitative analysis of the top-5 weak slices obtained using SWD-3 for the SETR semantic segmentation model trained and evaluated on Cityscapes dataset.

D Evaluation of YOLOv11m on EuroCity Persons dataset

We evaluate a publicly available YOLOv11m (Jocher & Qiu, 2024) model as our **DuT** containing 20.1 million parameters in the EuroCity Persons dataset (Braun et al., 2019). We separately analyse the “day” and “night” subsets provided by the dataset to enable a targeted investigation of the systematic weaknesses of the **DuT** model under varying lighting conditions. Given the challenging nature of the dataset, the YOLO model achieves a recall of only 0.42 on the day subset and 0.41 on the night subset, suggesting the presence of potential systematic failures. Consistent with our AD experiments, we exclude pedestrian instances occupying fewer than 3,000 pixels, as both human- and CLIP-based annotations exhibit low reliability for metadata generation on small instances due to data-induced uncertainty. Similar to table 3, we calculate the estimated precision and estimated recall for both subsets using human evaluation on 60 samples and present

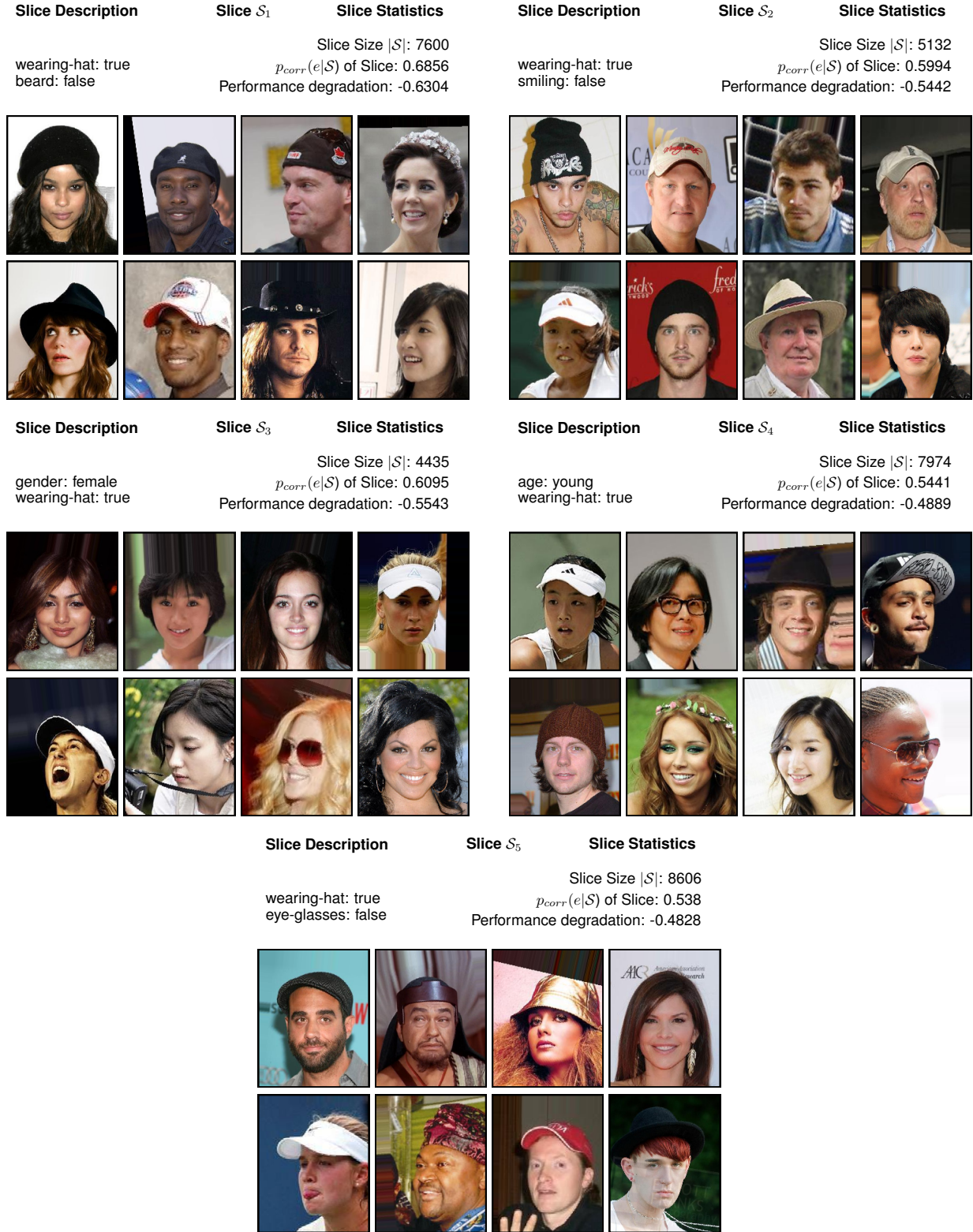


Figure 6: Samples from top-5 weak slices obtained using SWD-3 for the ViT-B-16 classification model trained on ImageNet21k and evaluated on the full CelebA dataset with metadata generated from CLIP using step 3 in algorithm 1. The statistics provide a quantitative evaluation of the entire slice. For qualitative evaluation, we provide some sample images from the slice.

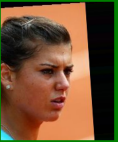







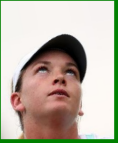









	Slice 1	Slice 2	Slice 3	Slice 4	Slice 5	Remaining Data
Slice Size $ S $	11726	2317	3344	2565	2307	180340
$p(e C)$ of Slice	0.6182	0.2486	0.0499	0.1053	0.1756	0.0146
Performance Degradation	-0.5622	-0.1926	0.0061	-0.0493	-0.1196	0.0414
						
						
						
						
						
						
						
						
Slice Descriptions	"a photo of the vocalist person"	"kate moss photo of a person"	"- bolivia photo of a person"	"judy garland photo of a person"	"a photo of a tennis person"	
	"a band photo of a person"	"a fashion photo of a person"	"a photo of a beauty person"	"a photo of judy garland person"	"a photo of a person playing playing tennis."	
	"a photo of the choreographer person"	"a photo of a person on runway."	"a woman photo of a person"	"a postwar photo of a person"	"a photo of a person playing tennis."	
	"a photo of a person or author."	"a photo of a person or model."	"a photo of a modeling person"	"a photo of a postwar person"	"a photo of a person clinching."	

Figure 7: Samples from top-5 weak slices of a ViT-B-16 classification model trained on ImageNet21k and evaluated on the full CelebA dataset (DOMINO). From the 8 samples in each slice, 4 are true positives (green outline) and 4 are false negatives (red outline).











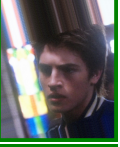




















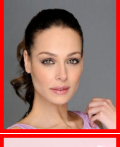
















	Slice 1	Slice 2	Slice 3	Slice 4	Slice 5	Remaining Data
Slice Size $ S $	4050	3930	1873	2498	3096	187152
$p(e C)$ of Slice	0.918	0.5542	0.6663	0.2894	0.1925	0.0151
Performance Degradation	-0.862	-0.4982	-0.6103	-0.2334	-0.1365	0.0409
						
						
						
						
						
						
						
						

Figure 8: Samples from top-5 weak slices of a ViT-B-16 classification model trained on ImageNet21k and evaluated on the full CelebA dataset (Spotlight). From the 8 samples in each slice, 4 are true positives (green outline) and 4 are false negatives (red outline). Spotlight does not provide automatic descriptions of the slices

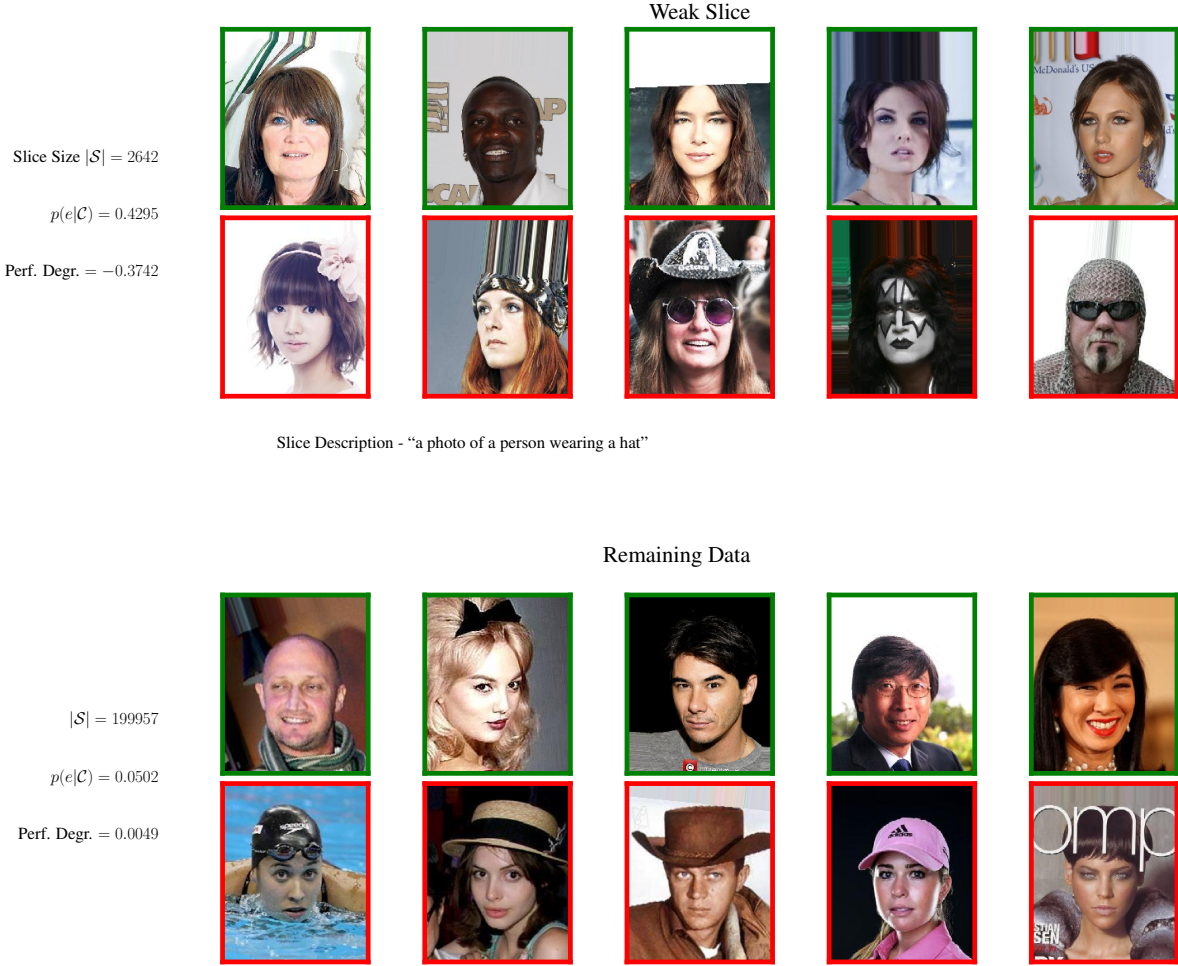


Figure 9: Samples from top-1 weak slices of a ViT-B-16 classification model trained on ImageNet21k and evaluated on the full CelebA dataset (SVM-FD). From the 10 samples in each slice, 5 are true positives (green outline) and 5 are false negatives (red outline). Unlike other SDMs, SVM-FD only outputs one weak slice.

Slice No.	$ \mathcal{S} $	$p_{\text{corr}}(e \mathcal{S})$	Avg. Perf. Degr.	Slice Description
\mathcal{S}_1	541	0.8663	-0.222	age: young
\mathcal{S}_2	510	0.8723	-0.228	age: young construction-worker: false
\mathcal{S}_3	405	0.8819	-0.2376	skin-color: dark cloth.-color: dark-color
\mathcal{S}_4	349	0.9095	-0.2652	age: young blurry: false
\mathcal{S}_5	173	1.00	-0.4602	age: young skin-color: dark

Table 9: Quantitative analysis of the top-5 weak slices obtained using SWD-3 for the Panoptic-FCN model trained and evaluated on RailSem19 dataset.

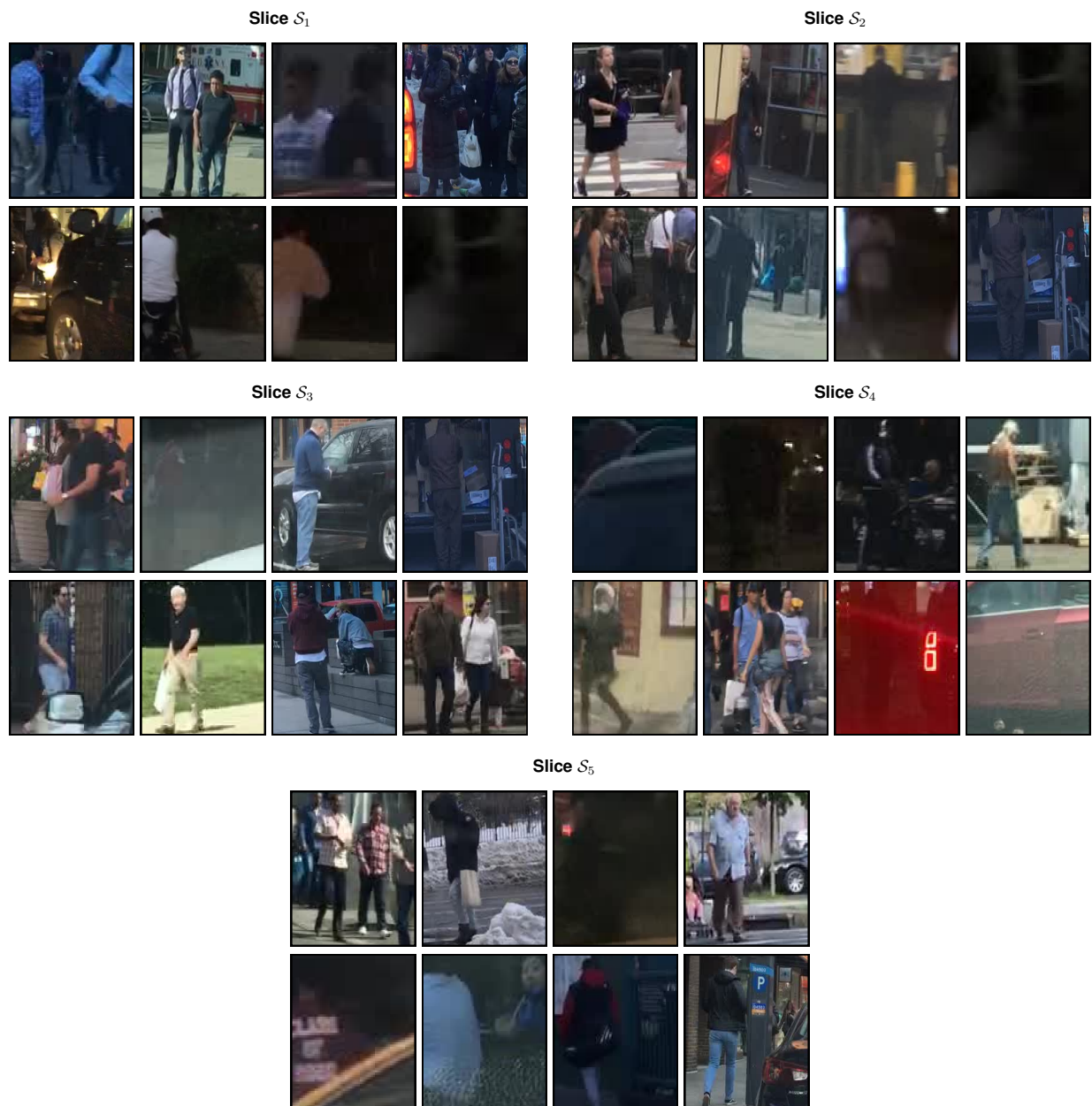


Figure 10: Samples from top-5 weak slices obtained using SWD-3 for the Faster R-CNN object detector trained and evaluated on BDD100k dataset.

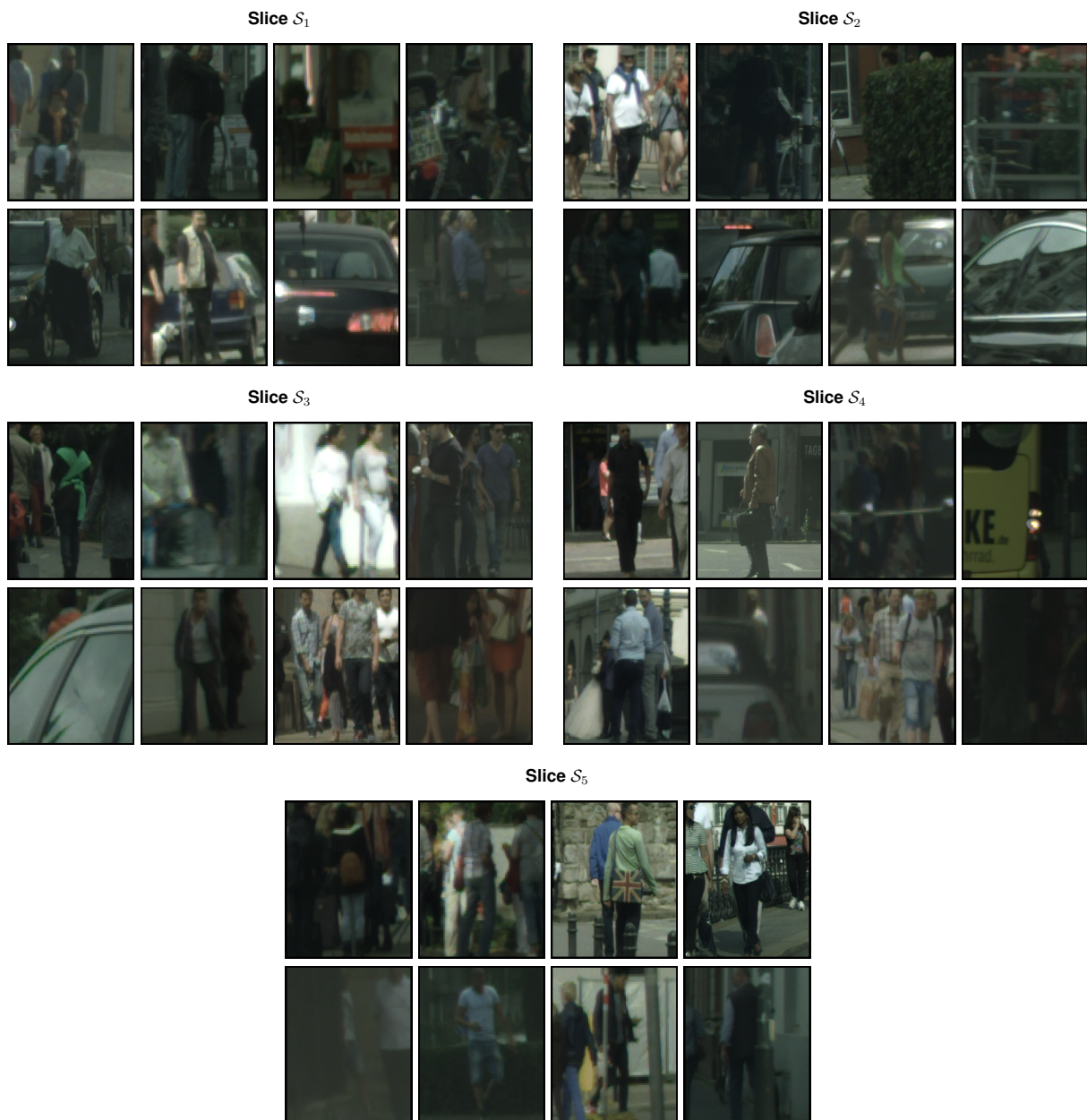


Figure 11: Samples from top-5 weak slices obtained using SWD-3 for the SETR semantic segmentation model trained and evaluated on Cityscapes dataset.

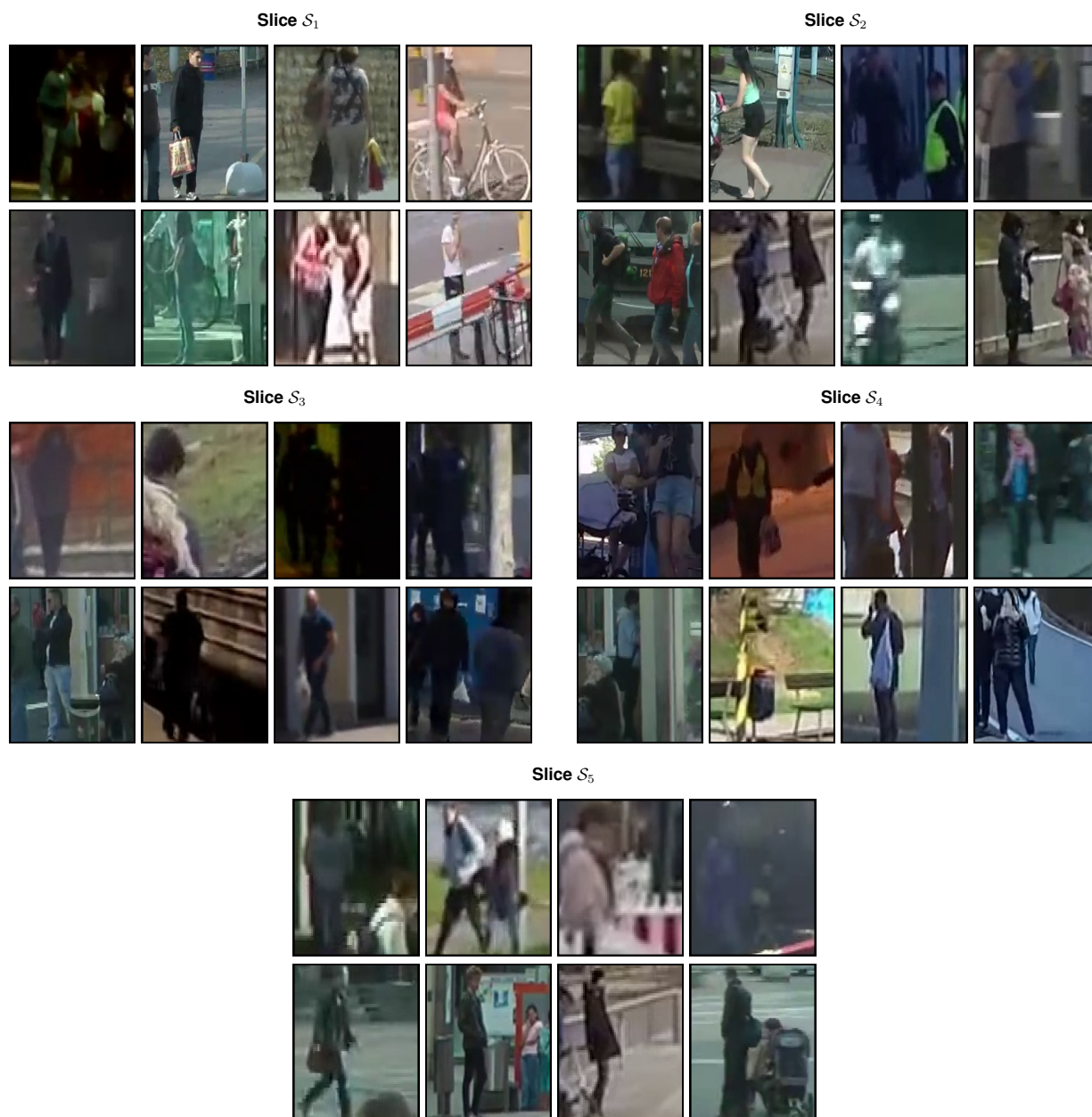


Figure 12: Samples from top-5 weak slices obtained using SWD-3 for the Panoptic-FCN model trained and evaluated on RailSem19 dataset.

the results in table 10. In samples where human evaluation is not clear, the benefit of the doubt is given to the CLIP model. This could, for instance, lead to higher precision in the night subset as it might be hard for the human labeler to also interpret some samples.

Based on the evaluation with SWD-3, we uncover that for the **DuT**, “occlusion” and its related sub-slices are the primary cause of systematic weaknesses in the day subset, compare table 11. For the night subset, table 12, pedestrians with potentially “dark” clothing or “dark” skin-color are semantics that lead to systematic weakness. The slice errors for all top-slices indicate significant performance degradation in the night and day subsets. Examples from the slices can be found in figures 13 and 14, respectively.

Sem. dim.	Attri.	Estimated Precision p_C		Estimated Recall r_C	
		EuroCity (Day)	EuroCity (Night)	EuroCity (Day)	EuroCity (Night)
Age	Adult	0.97 ± 0.02	0.92 ± 0.03	0.57 ± 0.02	0.54 ± 0.03
	Young	0.42 ± 0.06	0.63 ± 0.06	0.94 ± 0.06	0.93 ± 0.06
Gender	Female	0.89 ± 0.03	0.95 ± 0.03	0.88 ± 0.03	0.89 ± 0.03
	Male	0.92 ± 0.03	0.94 ± 0.03	0.91 ± 0.03	0.96 ± 0.03
Cloth.-color	Bright-color	0.70 ± 0.06	0.52 ± 0.06	0.26 ± 0.05	0.32 ± 0.06
	Dark-color	0.77 ± 0.05	0.90 ± 0.04	0.95 ± 0.05	0.95 ± 0.04
Skin-color	Dark	0.74 ± 0.06	0.73 ± 0.06	0.86 ± 0.06	0.87 ± 0.06
	White	0.97 ± 0.02	0.87 ± 0.04	0.91 ± 0.02	0.72 ± 0.04
Blurry	True	0.48 ± 0.06	0.53 ± 0.06	0.29 ± 0.06	0.40 ± 0.06
	False	0.79 ± 0.05	0.76 ± 0.05	0.89 ± 0.05	0.84 ± 0.05

Table 10: The estimated precision and recall using our proposed approach for evaluating the quality of the generated metadata for the EuroCity Persons dataset. Here, we provide the mean and $\sigma/2$, for n of 60, of the estimated precision and recall. Certain dimensions like occlusion are available as part of the datasets themselves. We do not perform human-evaluation for these dimensions but these are considered in the weak slice search.

Slice No.	$ \mathcal{S} $	$p_{\text{corr}}(e \mathcal{S})$	Avg. Perf. Degra.	Slice Description
\mathcal{S}_1	17227	0.6969	-0.373	occlusion: true blurry: false
\mathcal{S}_2	16732	0.6863	-0.3628	occlusion: true clothing-color: dark
\mathcal{S}_3	7215	0.6877	-0.3642	blurry: false skin-color: dark
\mathcal{S}_4	24484	0.6089	-0.2854	clothing-color: dark blurry: false
\mathcal{S}_5	5664	0.7045	-0.381	occlusion: true skin-color: dark

Table 11: Quantitative analysis of the top-5 weak slices obtained using SWD-3 for the YOLOv11m model evaluated on EuroCity Persons (day) dataset. For examples from the slices see fig. 13.

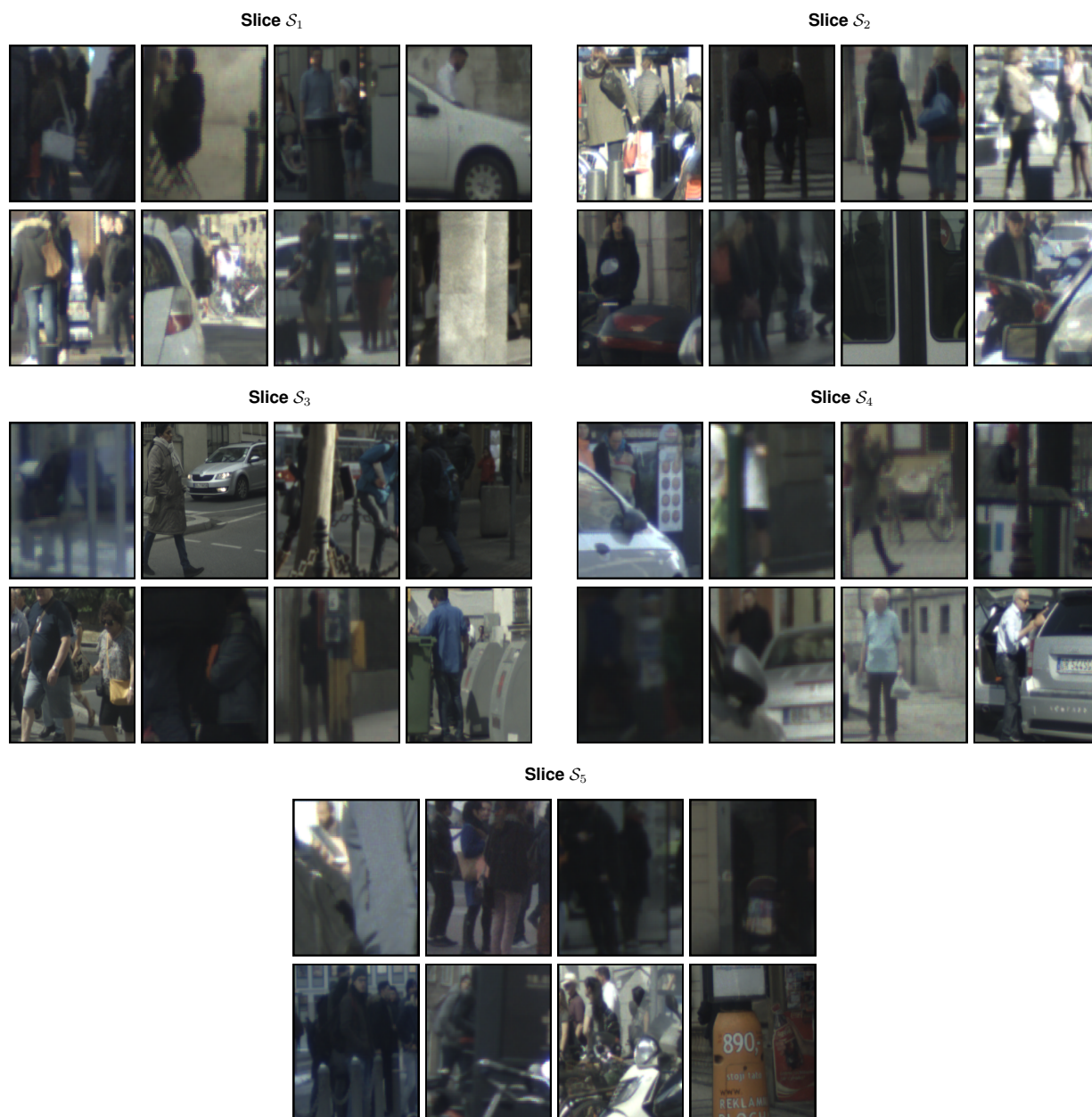


Figure 13: Samples from top-5 weak slices obtained using SWD-3 for the YOLOv11m model evaluated on EuroCity Persons (day) dataset. For the slice descriptions see the statistics in table 11.



Figure 14: Samples from top-5 weak slices obtained using SWD-3 for the YOLOv11m model evaluated on EuroCity Persons (night) dataset. For the slice descriptions see the statistics in table 12.

Slice No.	$ \mathcal{S} $	$p_{\text{corr}}(e \mathcal{S})$	Avg. Perf. Degra.	Slice Description
\mathcal{S}_1	7215	0.6877	-0.3079	skin-color: dark blurry: false
\mathcal{S}_2	24484	0.6089	-0.2291	clothing-color: dark blurry: false
\mathcal{S}_3	7091	0.6465	-0.2667	clothing-color: dark skin-color: dark
\mathcal{S}_4	6946	0.6188	-0.239	age: adult skin-color: dark
\mathcal{S}_5	8168	0.5608	-0.1809	age: young blurry: false

Table 12: Quantitative analysis of the top-5 weak slices obtained using SWD-3 for the YOLOv11m model evaluated on EuroCity Persons (night) dataset. For examples images from the slices see fig. 14.

E Comparison of CLIP and GPT-4o for metadata generation

We compare the metadata generation quality of CLIP with a more powerful model, GPT-4o (Hurst et al., 2024) in version ‘gpt-4o-2024-08-06’. As discussed, we expect newer models like GPT-4o to outperform CLIP due to advancements in training data quality, model architecture, and scale. Nevertheless, CLIP was chosen for this work due to its lightweight nature and ease of access, whereas large models such as GPT-4o may face scalability challenges. Following the evaluation strategy of (Gannamaneni et al., 2023), we assess both models on the full CelebA dataset (202,599 images) across relevant attributes for metadata generation. GPT-4o is accessed via the OpenAI API, with each image provided as input alongside a prompt that requests metadata in JSON format. The prompt template will be shared along with the experiment code. Table 13 reports the precision, recall, and F1 scores of the evaluations. CLIP outperforms GPT-4o only on the “Age” and “Pale Skin” attributes, while results for “Gender” and “Smiling” are comparable. For all other attributes, GPT-4o significantly outperforms CLIP. However, this improvement comes with considerable financial and computational costs (~ 170 million tokens), whereas CLIP offers negligible overhead in both respects. While manageable for the CelebA dataset, this might not scale for larger datasets. In either case, since the models do not have perfect precision, the error correction approach would need to be implemented to correctly identify the systematic weaknesses of a **DuT**.

Semantics	Attribute	Counts	CLIP			GPT-4o		
			Precision	Recall	F1 score	Precision	Recall	F1 score
Age	Young	156734	0.91	0.90	0.90	0.98	0.23	0.38
	Not-young	45865	0.67	0.68	0.68	0.27	0.99	0.43
Gender	Male	84434	0.99	0.99	0.99	0.99	0.99	0.99
	Not-male	118165	0.99	1.00	0.99	0.99	1.00	0.99
Pale-skin	Pale	8701	0.06	0.84	0.11	0.17	0.34	0.23
	Not-Pale	193898	0.98	0.39	0.56	0.97	0.92	0.95
Misc.	Eyeglasses	13193	0.50	0.94	0.65	0.94	0.98	0.96
	No eyeglasses	189406	1.00	0.93	0.96	1.00	1.00	1.00
	Hat	9818	0.52	0.87	0.65	0.71	0.96	0.82
	No Hat	192781	0.99	0.96	0.98	1.00	0.98	0.99
	Bald	4547	0.19	0.54	0.28	0.63	0.77	0.69
	Not Bald	198052	0.99	0.95	0.97	0.99	0.99	0.99
	Goatee	12716	0.21	0.62	0.31	0.57	0.62	0.60
	No Goatee	189883	0.97	0.84	0.90	0.97	0.97	0.97
	Beard	33441	0.31	0.33	0.32	0.95	0.44	0.60
	No Beard	169158	0.87	0.86	0.86	0.90	0.99	0.95
	Smiling	97669	0.84	0.85	0.85	0.90	0.91	0.91
	Not-smiling	104930	0.86	0.85	0.86	0.90	0.90	0.90

Table 13: The performance of CLIP in predicting different attributes on the celebrity images in the CelebA dataset similar to evaluations in (Gannamaneni et al., 2023). There are minor deviations between the results shown in this work and our CLIP results due to small changes in used prompts for CLIP text encoder.