Stratify or Die: Rethinking Data Splits in Image Segmentation

Naga Venkata Sai Jitin Jami[1,2] **Thomas Altstidl**[1] **Jonas Leo Mueller**[1] **Jindong Li**[1]
Dario Zanca[1] **Björn Eskofier**[1] **Heike Leutheuser**[2]

[1] Machine Learning and Data Analytics Lab, FAU Erlangen-Nürnberg, Germany
[2] Ambient Assisted Living and Medical Assistance Systems, Department of Computer Science, University of Bayreuth, Germany
{jitin.jami,thomas.r.altstidl,jonas.leo.mueller,jindong.li}@fau.de
{dario.zanca,bjoern.eskofier}@fau.de heike.leutheuser@uni-bayreuth.de

Abstract

Random splitting of datasets in image segmentation often leads to unrepresentative test sets, resulting in biased evaluations and poor model generalization. While stratified sampling has proven effective for addressing label distribution imbalance in classification tasks, extending these ideas to segmentation remains challenging due to the multi-label structure and class imbalance typically present in such data. Building on existing stratification concepts, we introduce Iterative Pixel Stratification (IPS), a straightforward, label-aware sampling method tailored for segmentation tasks. Additionally, we present Wasserstein-Driven Evolutionary Stratification (WDES), a novel genetic algorithm designed to minimize the Wasserstein distance, thereby optimizing the similarity of label distributions across dataset splits. We prove that WDES is globally optimal given enough generations. Using newly proposed statistical heterogeneity metrics, we evaluate both methods against random sampling and find that WDES consistently produces more representative splits. Applying WDES across diverse segmentation tasks, including street scenes, medical imaging, and satellite imagery, leads to lower performance variance and improved model evaluation. Our results also highlight the particular value of WDES in handling small, imbalanced, and low-diversity datasets, where conventional splitting strategies are most prone to bias.

1 Introduction

Image segmentation involves the assignment of a label to each pixel in an image, with applications in diverse fields such as autonomous driving [1–5], medical imaging [6–10], and satellite image analysis [11–15]. In a supervised learning setting, splitting the dataset into disjoint subsets for training and validation is common practice. However, following a random splitting strategy can result in class distribution shifts across subsets compared to the original dataset. This can lead to biased evaluations, as the test set is typically smaller than the training set and more affected by this phenomenon. Typically, such issues are addressed through stratification to guarantee a consistent distribution of class labels across subsets [16] but this is not straightforward for image segmentation datasets [17–21]. The fundamental unit, an image, is not a sample with a single class but a sample with pixels belonging to many classes. Stratification strategies for similarly complex multi-label datasets [22] could provide interesting solutions that can be applied to segmentation datasets. Strategies relevant to multi-label datasets have been proposed and investigated in various forms ranging from iterative strategies [23–25], greedy search [26, 27], and genetic algorithms [28].These methods do not directly apply to the problem of segmentation, as multi-label datasets have the complexity of multiple positive

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

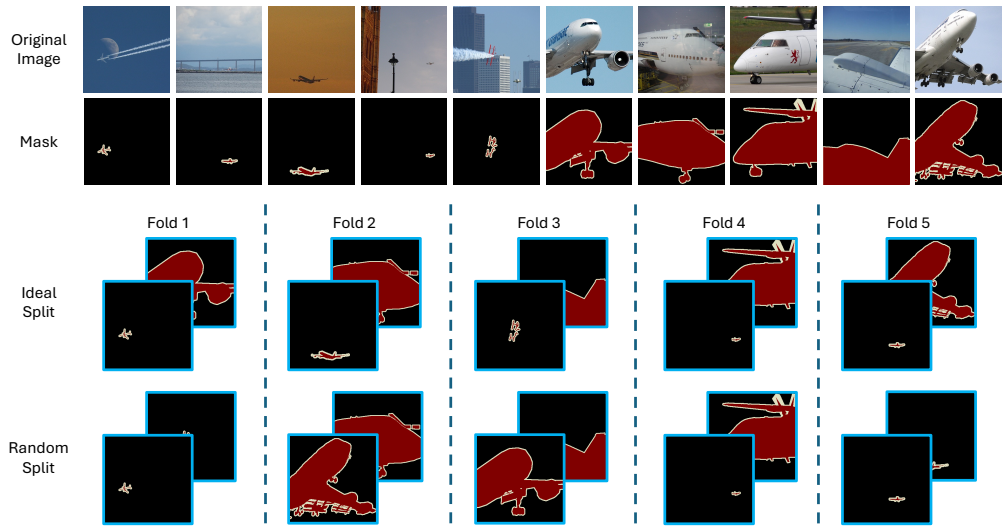

Figure 1: Representative examples of *Plane* images from PascalVOC [17]. The illustration shows an ideal split with balanced class distribution across folds which is not ensured by random splitting. Such balance is essential to avoid bias and ensure reliable model evaluation.

classes for each sample but do not have the added complexity of proportional representation of these classes in the sample. Segmentation datasets present a unique challenge in this regard, complicating the task of stratification.

Segmentation datasets are often small due to high annotation costs and exhibit imbalanced class distributions due to label diversity. As dataset size decreases, inconsistent class distributions in subsets increase, hindering generalization [29]. This issue is critical in imbalanced datasets, where some classes may be absent from splits, making performance metrics like F1-score and IoU unmeasurable. Such missing data undermines the reliability of model evaluation, especially in K-fold cross-validation [30], where unrepresentative subsets affect performance variance.

Motivated by these challenges, we propose and investigate two stratification strategies: *Iterative Pixel Stratification (IPS)* and *Wasserstein-Driven Evolutionary Stratification (WDES)*[1]. These methods are designed to assign samples to subsets, or folds, for a train-test split scenario (two folds) or a K-fold cross-validation scenario (K folds), taking into account the label composition of the sample. IPS extends the Iterative Stratification (IS) algorithm proposed in literature [23], assigning samples to folds using a greedy approach to meet predefined class presence requirements measured by the number of pixels in each fold. It prioritizes assigning samples containing rare classes to folds, progressively addressing those with increasing ubiquity. In contrast, WDES employs an evolutionary algorithm akin to Evosplit [28], where a population of potential fold assignments (individuals) is evaluated simultaneously. The fittest individuals undergo selective crossover and mutations for further refinement. The fundamental distinction between these two methods lies in their approach to fold assignment and evaluation. The iterative nature of IPS confines it to a sequential decision-making process and is limited to a single chain of iterations. Conversely, the genetic algorithm in WDES evaluates multiple solutions independently by maintaining a population, enabling it to explore a broader solution space. The flexibility of WDES allows us to choose a better objective function to be minimized. We choose the Wasserstein distance between the class distribution in the fold and the overall dataset as it indicates the similarity between these distributions. Hence, minimizing the distance reduces dissimilarity between the subsets and the dataset.

We evaluate IPS, WDES, and random splitting across five benchmark segmentation datasets. Our evaluation begins by analyzing the consistency of folds generated through repeated experiments. We

[1]Implementation available at: `https://github.com/jitinjami/SemanticStratification`

further assess the impact of targeted stratification versus random splitting by examining the standard deviation of accuracy, F1-score, and Intersection over Union (IoU) across 10-fold cross-validation. A lower deviation indicates a more reliable assessment of model performance. Empirical results show that WDES consistently achieves the highest quality splits. In contrast, IPS under-performs relative to WDES, suggesting that a sequential strategy may be suboptimal for segmentation tasks. Notably, WDES also demonstrates better consistency over the other two for evaluating model performance on low-entropy datasets, where class distributions are highly imbalanced or concentrated. Overall, our findings highlight the importance of stratification tailored to structured outputs and advocate for the adoption of principled splitting methods in segmentation tasks.

2 Related Work

Existing methodologies for stratifying complex datasets trace their origins to Iterative Stratification (IS) by [23]. In their work, the authors addressed the intricacies of multi-label datasets, proposing a greedy algorithm that iteratively assigns samples to folds based on the required frequency of positive labels within each fold. Fold quality is evaluated by comparing the Label Distribution (LD) metric. LD is the difference between the proportion of samples belonging to a given class within a fold and the corresponding proportion in the overall dataset

The partitioning method based on stratified random sampling (PMBSRS) by [24] employs a similar iterative approach but incorporates a pre-sorting step. This method clusters samples with similar label sets before assigning them equally to folds, thereby enhancing the homogeneity of label distribution across folds. Building upon the work of [23], authors of [25] introduced Second Order Iterative Stratification (SOIS), which considers second-order label relationships. Instead of focusing on individual labels, SOIS accounts for label pairs to determine the demand within each fold, thereby capturing more complex interactions between labels.

Other strategies approach stratification by starting from an initial assignment and iteratively refining it. The Stratified Sampling (SS) method proposed by [26] pre-assigns samples to folds and evaluates them based on deviations from the ideal label distribution. Samples contributing to the highest deviation subset are shuffled between folds to minimize discrepancies. This approach was designed with large-scale XML datasets in mind [31]. A similar split-and-shuffle approach was adopted by [27] but from a label-centric perspective. They developed a scoring mechanism to assess fold quality relative to each class, shuffling samples associated with the class exhibiting the poorest quality before recalculating scores. This method demonstrated effectiveness on large Gene Ontology datasets.

Finally, Evosplit, proposed by [28], leverages a genetic algorithm to address the challenges outlined by [23] and [25]. This method generates a population of fold assignments, evaluates their fitness, performs crossover and mutation, and iterates over generations to arrive at an optimal solution. Fitness is determined using Label Distribution (LD) from [23] and Label Pair Distribution (LPD) from [25].

While these methods offer valuable strategies for stratifying multi-label datasets, they do not directly extend to image segmentation. In segmentation, each sample contains a dense array of pixel-wise labels rather than a simple set of associated classes, and the proportion of pixels per class varies significantly across images. This makes stratification more complex: it is not enough to ensure the presence of a class in a fold; one must also consider how extensively each class is represented at the pixel level. Therefore, segmentation-specific stratification requires reformulating both the problem and the evaluation criteria. In the next section, we formalize the segmentation stratification task and introduce two new approaches, Iterative Pixel Stratification (IPS) and Wasserstein-Driven Evolutionary Stratification (WDES), which aim to create representative folds by explicitly accounting for pixel-level label distributions.

3 Stratification algorithms

Consider an image segmentation dataset $D = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$ with N samples, where $\mathbf{x}_n$ denotes the input images and $\mathbf{y}_n$ the corresponding label masks. Let the total number of pixels across all samples be P, and the number of semantic classes be C. The objective is to partition D into K disjoint subsets, denoted $\mathbf{S} = (S^1, \ldots, S^K)$, according to a target proportion vector $\mathbf{r} = (r^1, \ldots, r^K)$, where r^k specifies the desired proportion of samples in fold k.

Finding such a partition that preserves the global class distribution across folds is NP-Hard because it can be reduced to a multi-way partition problem [32]. It is analogous to dividing a multi-set of weighted elements into K subsets with balanced sums; except here, the "weights" are pixel counts per class and the balance is across multiple classes simultaneously

Let N_c be the number of samples in the dataset that contain class c, and N_c^k the required number of such samples in fold k. Similarly, let P^n be the total number of pixels in sample n, and P_c^n the number of pixels of class c in sample n. The total number of pixels of class c in the dataset is P_c, and the desired number in fold k is P_c^k. These quantities are central to the evaluation of fold quality and to guiding the stratification process.

3.1 Similarity measures

We first introduce similarity measures calculated on the subsets to assess the quality of the stratified folds $\mathbf{S}$. These statistical properties of the subsets aim to quantify: (1) how well each fold adhere to the target sample proportions defined by $\mathbf{r}$; (2) how closely the class-wise pixel proportions in each subset match those in the full dataset; (3) the overall similarity between fold-level and global label distributions using distributional distance.

Sample Distribution (*SD*) With this measure, we aim to assess the deviation of the number of samples in each subset from the required number of samples as defined by the proportion vector $\mathbf{r}$. For example, if a dataset of 10 samples is to be equally divided but the subsets contain 4 and 6 samples respectively, the value SD is 1. It is calculated as the average across folds of the deviations from the desired number of samples in each subset. A low SD indicates that the number of samples per fold closely matches the intended proportions.

$$\text{SD} = \frac{1}{K} \sum_{k=1}^{K} ||S^k| - N^k| \tag{1}$$

where $N^k = r^k \cdot N$ is the expected number of samples in fold k.

Pixel Label Distribution (*PLD*) This measure evaluates how closely the proportion of pixels of each class within each subset matches the proportion of that class in the entire dataset. For each class c, it calculates the absolute difference between the ratio of pixels of that label in each subset k and the ratio of pixels of that label in the whole dataset D. It then averages these results across all labels. This measure is inspired by Label Distribution (LD) in [23]. Lower PLD values indicate better preservation of class proportions at the pixel level.

$$\text{PLD} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{K} \sum_{k=1}^{K} \left| \frac{P_c^k}{P_c - P_c^k} - \frac{P_c}{P - P_c} \right| \right) \tag{2}$$

An additional similarity measure examining class co-occurrence relationships is provided in Appendix H, where we extend this evaluation to capture second-order interactions between classes using the Pixel Label Pair Distribution (PLPD).

Label Wasserstein Distance (*LWD*) This measure uses the Wasserstein Distance to quantify the dissimilarity between the cumulative class distributions in each fold and that of the entire dataset. By calculating this distance for each subset and averaging the results, LWD provides an indication of how closely the pixel-level class distributions in the subsets resemble that of the original dataset. Let F_c and F_c^k denote the cumulative pixel distributions up to class c in the dataset and in fold k, respectively:

$$F_c = \sum_{i=1}^{c} P_i, \qquad F_c^k = \sum_{i=1}^{c} P_i^k$$

Then the Label Wasserstein Distance is defined as:

$$\text{LWD} = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} \left| F_c - F_c^k \right| \tag{3}$$

To complement the Wasserstein-based similarity assessment, Appendix H presents further similarity assessments using alternative distributional measures, namely the Linear-kernel Maximum Mean Discrepancy (L-MMD) and the Linear Energy Distance (L-ED).

3.2 Iterative Pixel Stratification (*IPS*)

The motivation behind Iterative Pixel Stratification (IPS) stems from Iterative Stratification (IS) [23], which aims to distribute samples with specific positive labels proportionally across folds. This distribution is guided by two key properties within each fold: the number of samples required and the number of positive label instances required. To achieve this, IS starts by identifying the rarest label among the unassigned samples, determines which fold most requires positive instances of it and assigns samples that have positive instances of this label to that fold. In cases where multiple folds have equal demand, the fold with the fewest total samples is prioritized. Once a sample is assigned to a fold, it is removed from the pool of unassigned samples, and the demand for all positive labels associated with that sample is reduced in the fold. This iterative process continues, with the demand for specific positive labels decreasing as samples are assigned. The process concludes once all samples have been allocated, typically after C iterations.

IPS differs from IS in one key way: the label demand in folds is determined by the number of pixels required, not by the positive instances of that label. This also means that when a sample is assigned to a fold, the desire for all labels is reduced by the number of pixels of every label in that sample. As a result, IPS emphasizes achieving pixel-level balance over maintaining sample count proportionality. Note that IPS is implemented by adapting Iterative Stratification without additional optimizations thereby serving as a straightforward baseline. The pseudo-code for IPS algorithm is presented in Appendix B. An alternative iterative stratification approach is presented in Appendix G.

3.3 Wasserstein-Driven Evolutionary Stratification (*WDES*)

WDES formulates the stratification problem as an optimization task and solves it using a genetic algorithm. Each candidate solution, or *individual*, represents an assignment of samples to folds. Initially, a population of individuals is generated by randomly assigning samples to folds according to the target proportion vector $\mathbf{r}$. Each *gene* (element) in an individual encodes the fold assignment of a specific sample.

The evolution of individuals is driven by crossover and mutation. First, we evaluate fitness using LWD (Eq. 3) and perform tournament selection [33]. Crossover occurs with a fixed probability, exchanging gene segments between individuals. A correction step follows to maintain target proportions across folds, reassigning samples if necessary. Mutation swaps assignments of randomly selected pairs, preserving fold proportions and adding diversity. This cycle repeats for a set number of generations, with the fittest individual (i.e., with the lowest LWD) selected as the final stratification.

We use the LWD as the fitness function to measure class distribution similarity. The proportionality requirement is managed through the initial population and crossover design, ensuring the genetic algorithm optimizes class distribution while maintaining proportionality across folds for a balanced stratification.

3.4 On the convergence of stratification algorithms

For large multi-label datasets, as the dataset size grows, random splits asymptotically preserve the marginal label frequencies in each fold due to the law of large numbers. We provided a mathematical proof of this property in Appendix A.1. This property makes random stratification a practical choice for very large datasets, where computational efficiency and simplicity are preferable.

For smaller datasets, where N is finite, WDES provides strong guarantees of approaching the optimal stratification as the number of generations and population size increase.

Proposition 3.1 (Convergence Rate of WDES to Empirical Optimum)**.** *Let $\mathcal{F}_{M,G} \subset \mathcal{F}$ denote the set of stratifications explored by the WDES algorithm after G generations with a population size M. Then the expected suboptimality of the final stratification $\mathcal{S}_{WDES} \in \mathcal{F}_{M,G}$ satisfies:*

$$\mathbb{E}\left[L(\mathcal{S}_{WDES})\right] - L^*_{\mathcal{F}_{M,G}} \leq \mathcal{O}\left(\frac{1}{MG}\right).$$

where $L^*_{\mathcal{F}_{M,G}}$ denotes the optimal Wasserstein score in the empirical subset.

Proof. See Appendix A.2. □

The above results shows that the expected quality of the WDES stratification improves at a rate of O(1/MG), meaning that as the number of generations and population size increase, the WDES algorithm converges toward the global optimum. This makes WDES a particularly effective choice for small datasets, where the algorithm can afford the additional computational cost required to reach an optimal stratification, providing guarantees of high-quality splits as the search progresses.

4 Experiments

We conduct a comparative analysis of three stratification strategies: a) random sampling using KFold from the `scikit-learn` library [34], b) Iterative Pixel Stratification, and c) Wasserstein-Driven Evolutionary Stratification implemented using the `deap` library [35].

To achieve this, we start by testing these algorithms on the datasets outlined in Section 4.1 to calculate the statistical similarity measures (described in Section 3.1). For all the algorithms, we consider the number of pixels belonging to each class in the provided mask of every sample to guide the stratification procedure. The applicable parameters used for WDES are outlined in Appendix D. We perform stratification with datasets randomly shuffled five times and average the calculated similarity measures.

Following this, we perform 10-fold cross-validation tests with random splitting and the targeted stratification strategies. For every fold, we train a UNet [6] with a `resnet34` encoder from [36] for 50 epochs to perform segmentation on the images. The training process employs a learning rate of `2e-4` and utilizes the `Adam` optimizer with `DiceLoss`. For PascalVOC, `CELoss` is used instead, with a learning rate of `1e-4` and trained for 100 epochs. All experiments are conducted on a single Nvidia A100 graphics card with 40GB of VRAM, without distributed training, ensuring consistent and comparable results across the different stratification methods. We evaluate the stratification methods by measuring the variance in model performance across folds, to assess whether targeted stratification provides more stable and reliable evaluations.

4.1 Datasets

We select datasets spanning four major application domains: autonomous driving/street scenes, medical images, satellite imagery, and general-purpose datasets. For autonomous driving, we use Cityscapes [21] and CamVid [18]. In the medical domain, we consider EndoVis2018, a robotic scene segmentation dataset from MICCAI 2018 [19]. For satellite imagery, we use the LoveDA dataset [20], which focuses on land-cover segmentation in urban and rural locations. For general-purpose segmentation tasks, we include PascalVOC 2012 [17].

To characterize the complexity of each dataset, we define several key properties. Class Cardinality (CC) refers to the average number of classes present in a sample. Class Ubiquity (CU) captures the average number of samples in which each class appears, indicating how frequently each class occurs across the dataset. Average Imbalance Ratio (AIR) is the mean of class-wise imbalance ratios, where the imbalance ratio for a given class is the ratio of pixels in the most frequent class to those in the class under consideration. Entropy quantifies the diversity in class proportions; higher entropy implies a more balanced distribution of classes, whereas lower entropy indicates imbalance. These metrics provide a nuanced understanding of the datasets' structural diversity and segmentation challenges. The properties of all datasets are summarized in Table 1, sorted by entropy.

Table 1: Datasets and properties

Dataset	N Images	N Classes	CC	CU	AIR	Entropy
PascalVOC [17]	2,913	21	2.48	344	79	1.82
CamVid [18]	701	32	17.44	370	7.56E+06	1.94
EndoVis [19]	2,235	12	6	1,117	1,492	2.38
LoveDA [20]	2,522	8	4.97	1568	4.5	2.61
CityScapes [21]	2,974	35	16.6	1,412	467	3.17

Table 2: Comparison of stratification methods using statistical similarity measures across datasets. WDES yields folds that better reflect the overall distribution, as indicated by PLD and LWD. The advantage narrows for datasets with higher class cardinality and lower ubiquity.

Dataset	Method	SD		PLD $(\times 10^{-5})$		LWD $(\times 10)$	
		Mean	Std	Mean	Std	Mean	Std
PascalVOC	Random	**0.42**	**0.00**	955	137	75.0	2.62
	IPS	10.28	2.46	733	244	53.3	1.20
	WDES	**0.42**	**0.00**	**456**	**77.9**	**51.4**	**1.63**
Camvid	Random	**0.00**	**0.00**	63.4	3.73	183	5.08
	IPS	6.68	1.18	76.6	5.30	209	8.44
	WDES	**0.00**	**0.00**	**36.7**	**4.47**	**138**	**7.84**
EndoVis	Random	**0.50**	**0.00**	609	81.4	661	35.5
	IPS	11.42	1.05	769	12.2	764	75.2
	WDES	**0.50**	**0.00**	**208**	**23.8**	**399**	**10.4**
LoveDA	Random	**0.32**	**0.00**	1110	109	1080	88.4
	IPS	8.06	1.15	1090	207	1100	87.4
	WDES	**0.32**	**0.00**	**377**	**31.4**	**635**	**16.0**
Cityscapes	Random	**0.50**	**0.00**	126	15.4	304	16.5
	IPS	11.54	2.69	148	17.4	324	22.2
	WDES	**0.50**	**0.00**	**67**	**3.44**	**217**	**5.17**
Ranking	Random	**1**		2.4		2.2	
	IPS	2		2.6		2.8	
	WDES	**1**		**1**		**1**	

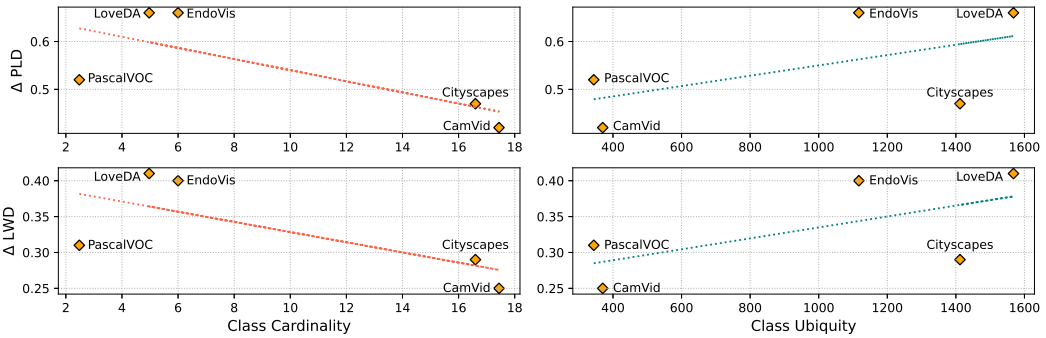

Figure 2: Relative improvement of WDES over random splitting in PLD and LWD. Performance gap decreases with increasing class cardinality, but WDES maintains an edge in datasets with high class ubiquity.

Table 3: Average segmentation performance of UNet across stratification methods in 10-fold cross-validation. Standard deviations in Accuracy, F1, and IoU are used to assess consistency. WDES performs best in low-entropy datasets, while random splitting becomes competitive as entropy increases.

Dataset	Method	Accuracy		F1		IoU	
		Mean	Std ($\times 10^{-3}$)	Mean	Std ($\times 10^{-3}$)	Mean	Std ($\times 10^{-3}$)
PascalVOC	Random	0.76	20.6	0.58	32.5	0.44	34.3
	IPS	0.76	14.5	0.58	30.3	0.44	31.6
	WDES	0.75	**11.0**	0.57	**24.0**	0.43	**24.2**
Camvid	Random	0.94	0.68	0.91	1.16	0.89	1.17
	IPS	0.94	0.74	0.91	1.35	0.89	1.27
	WDES	0.94	**0.67**	0.91	**1.09**	0.89	**1.06**
EndoVis	Random	0.94	19.4	0.86	27.0	0.80	28.1
	IPS	0.94	14.1	0.87	29.3	0.79	30.7
	WDES	0.94	**13.0**	0.85	**17.9**	0.79	**18.9**
LoveDA	Random	0.88	**6.63**	0.80	**6.39**	0.69	**8.38**
	IPS	0.88	7.39	0.80	9.29	0.69	10.7
	WDES	0.88	7.49	0.80	9.86	0.69	11.0
Cityscapes	Random	0.79	11.2	0.61	**9.12**	0.50	**8.64**
	IPS	0.79	**10.3**	0.61	14.9	0.50	15.2
	WDES	0.79	11.2	0.61	14.4	0.50	12.8
Ranking	Random		2.4		1.8		1.8
	IPS		2		2.6		2.6
	WDES		**1.6**		**1.6**		**1.6**

5 Results and Discussion

5.1 Distribution of Labels and Samples

Table 2 presents the statistical similarity measures calculated for the subsets generated across the five datasets. For each measure and dataset, we report the mean over five runs, highlighting the best-performing method in bold. An overall average rank is also computed for each method by assigning rank 3 to the method with the lowest mean and rank 1 to the one with the highest.

Our findings indicate that WDES consistently achieves superior performance in terms of PLD and LWD across all datasets, underscoring its effectiveness in preserving label proportions and label distributions within the stratified subsets. Although IPS was explicitly designed to address proportionality in class presence, WDES outperforms it not only in LWD (which it directly optimizes) but also in PLD. This reinforces the perspective that stratification for image segmentation tasks benefits more from minimizing distributional dissimilarity than from simply ensuring class proportions.

WDES also exhibits favorable SD scores, which follow directly from its design. Because WDES explicitly enforces predefined sample proportions across folds, it naturally results in balanced sample counts. This behavior is similarly observed in random splitting, which allocates samples proportionally across folds without replacement.

Another noteworthy pattern emerges in relation to dataset complexity. As class cardinality increases, the performance gap between WDES and random splitting diminishes for both PLD and LWD. This trend suggests that when more classes are present per sample, the benefits of targeted stratification decrease due to the inherent balancing effect of high cardinality. In contrast, as class ubiquity increases, indicating that classes are more commonly present across the dataset, the superiority of WDES becomes more pronounced. This is expected, as higher ubiquity leads to more overlap between classes and samples, thereby amplifying the benefits of optimization-based stratification. These trends are illustrated in Figure 2, which presents the changes in PLD and LWD differences relative to class cardinality and ubiquity using trend lines.

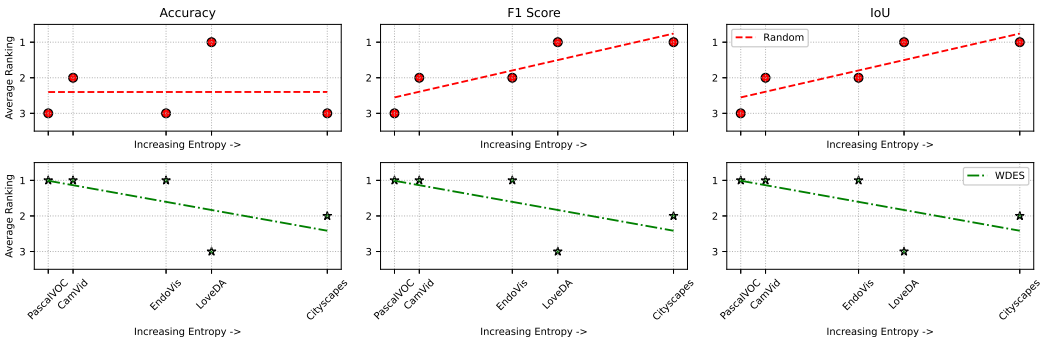

Figure 3: Stratification method rankings by performance variance across datasets. WDES ranks highest in low-entropy settings; random splitting improves with higher entropy, reducing the benefit of targeted stratification.

Table 4: Average segmentation performance of UNet for underrepresented classes (*bicycle*, *boat*, and *potted plant*) in PascalVOC under 10-fold cross-validation. Standard deviations in Accuracy, F1 Score, and IoU are reported to assess consistency. WDES yields the most stable results across all three rare classes, indicating improved fold balance for low-frequency categories.

Class	Method	Accuracy		F1		IoU	
		Mean	Std	Mean	Std	Mean	Std
Bicycle	Random	0.62	0.04	0.31	0.08	0.19	0.06
	IPS	0.62	0.04	0.31	0.09	0.19	0.06
	WDES	0.62	**0.03**	0.31	**0.07**	0.19	**0.05**
Boat	Random	0.72	0.05	0.53	0.09	0.37	0.08
	IPS	0.70	0.06	0.49	0.14	0.33	0.12
	WDES	0.72	**0.04**	0.55	**0.08**	0.38	**0.08**
Potted Plant	Random	0.71	0.08	0.47	0.16	0.32	0.14
	IPS	0.71	0.09	0.51	0.16	0.36	0.14
	WDES	0.70	**0.07**	0.47	**0.12**	0.31	**0.10**

5.2 Variance of Performance

Table 3 presents the results of training a UNet model on each dataset using 10-fold cross-validation. For each fold, we evaluate performance using Accuracy, F1 Score, and Intersection over Union (IoU), and report the average across the ten experiments. To highlight the robustness of each stratification strategy, we identify the method with the lowest standard deviation for each metric. Rankings are assigned from 1 to 3, with 1 indicating the lowest variance in performance and thus better fold consistency.

Our analysis shows that WDES achieves the best average rank in standard deviation across Accuracy, F1 Score, and IoU. This indicates that WDES produces splits that are more consistent with each other and better reflect the overall data distribution, leading to more stable model evaluation. While we cannot distinguish whether the reduced variance stems from the training or test set, the smaller size of the test set suggests it is more affected by sampling variability. Thus, the improvements in test set consistency are likely the main driver of the reduced standard deviation. This makes the average cross-validation score under WDES a more reliable estimate of a model's true generalization performance. We leave empirical validation of this to future work.

To further examine whether these improvements extend to underrepresented categories, we analyzed the segmentation performance for three rare classes in PascalVOC: *bicycle*, *boat*, and *potted plant*. As shown in Table 4, WDES consistently yields lower standard deviation in Accuracy, F1 Score, and IoU for these classes compared to random and IPS-based splits. This mirrors the global trend

observed in our main results, suggesting that WDES not only improves stability across folds overall but also enhances evaluation consistency for rare classes in low-entropy datasets.

As we move from low-entropy to high-entropy datasets, where label distributions are inherently more balanced and diverse, the advantage of WDES diminishes. The performance of random splitting improves and can even outperform stratified approaches. In these cases, the class distributions are already relatively uniform across samples, making targeted stratification less critical. This trend is reflected in the increasing rank of random splitting and the decline in WDES performance with increasing entropy, as illustrated in Figure 3.

While IPS was designed to be a label-aware stratification method, it ranks lowest on average across accuracy, F1, and IoU deviation. A potential reason lies in its single-pass, greedy allocation strategy, which reduces label demands in a fold as soon as a sample is assigned. This mechanism introduces a path dependency that prevents the algorithm from reassessing earlier decisions or accounting for broader distributional goals. Furthermore, although IPS considers pixel-level proportions, its strategy is primarily focused on satisfying per-label pixel quotas without evaluating the resulting inter-label distribution in each fold. As a result, by prioritizing rare labels first and assigning samples based solely on immediate label demand, it may overlook more nuanced strategies for dividing the dataset, such as balancing co-occurring label groups, preserving contextual diversity, or accounting for interdependencies between frequent and infrequent classes.

6 Conclusion and Limitations

In this paper, we introduced WDES, a stratification method for image segmentation tasks based on a genetic algorithm that minimizes the Wasserstein distance between label distributions. We provide a theoretical guarantee (Appendix A.2) that WDES converges to a globally optimal stratification given sufficient generations. We empirically compared its performance against iterative (IPS) and random sampling across several datasets and evaluation criteria. WDES outperforms both iterative and random methods, achieving the best average rank in accuracy, F1 score, and IoU variance in segmentation tasks, particularly for low-entropy datasets. Conversely, random stratification shows lower variance in high-entropy datasets, indicating that the benefits of targeted stratification diminish as class distributions become more uniform.

We emphasize the importance of stratification in cross-validation and train-test splits for image segmentation, particularly to avoid underrepresenting certain test sets and biasing the evaluation. This issue is especially critical for rare or underrepresented classes and when working with small datasets. Because final performance scores in cross-validation are typically computed as the mean of fold-wise means, the effects of imbalanced splits are not mitigated by the repeated nature of the process. While WDES improves upon simpler strategies, it is not a universal solution. Its reliance on minimizing Wasserstein distance assumes that distributional similarity alone ensures balanced splits, potentially overlooking other factors such as spatial structure or class co-occurrence. Future work should explore alternative similarity measures and more comprehensive modeling of dataset composition to improve stratification in segmentation tasks.

References

[1] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 695–714. Springer, 2020.

[2] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

[3] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3349–3364, 2020.

[4] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 752–761. IEEE, 2023.

[5] Serdar Erişen. Sernet-former: Segmentation by efficient-resnet with attention-boosting gates and attention-fusion networks. In *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, pages 1–6. IEEE, 2024.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[7] Debesh Jha, Michael A Riegler, Dag Johansen, Pål Halvorsen, and Håvard D Johansen. Doubleu-net: A deep convolutional neural network for medical image segmentation. In *2020 IEEE 33rd International symposium on computer-based medical systems (CBMS)*, pages 558–564. IEEE, 2020.

[8] Razvan-Gabriel Dumitru, Darius Peteleaza, and Catalin Craciun. Using duck-net for polyp image segmentation. *Scientific reports*, 13(1):9803, 2023.

[9] Meng Wei, Charlie Budd, Luis C Garcia-Peraza-Herrera, Reuben Dorent, Miaojing Shi, and Tom Vercauteren. Segmatch: A semi-supervised learning method for surgical instrument segmentation. *arXiv preprint arXiv:2308.05232*, 2023.

[10] Ioannis A Vezakis, Konstantinos Georgas, Dimitrios Fotiadis, and George K Matsopoulos. Effisegnet: Gastrointestinal polyp segmentation through a pre-trained efficientnet-based network with a simplified decoder. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–4. IEEE, 2024.

[11] Libo Wang, Rui Li, Chenxi Duan, Ce Zhang, Xiaoliang Meng, and Shenghui Fang. A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[12] Taisei Hanyu, Kashu Yamazaki, Minh Tran, Roy A McCann, Haitao Liao, Chase Rainwater, Meredith Adkins, Jackson Cothren, and Ngan Le. Aerialformer: Multi-resolution transformer for aerial image segmentation. *Remote Sensing*, 16(16):2930, 2024.

[13] Gyutae Hwang, Jiwoo Jeong, and Sang Jun Lee. Sfa-net: Semantic feature adjustment network for remote sensing image segmentation. *Remote Sensing*, 16(17):3278, 2024.

[14] Ivica Dimitrovski, Vlatko Spasev, Suzana Loshkovska, and Ivan Kitanovski. U-net ensemble for enhanced semantic segmentation in remote sensing imagery. *Remote Sensing*, 16(12):2077, 2024.

[15] Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, and Peter M Atkinson. Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190: 196–214, 2022.

[16] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[18] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern recognition letters*, 30(2):88–97, 2009.

[19] Max Allan, Satoshi Kondo, Sebastian Bodenstedt, Stefan Leger, Rahim Kadkhodamohammadi, Imanol Luengo, Felix Fuentes, Evangello Flouty, Ahmed Mohammed, Marius Pedersen, et al. 2018 robotic scene segmentation challenge. *arXiv preprint arXiv:2001.11190*, 2020.

[20] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/4e732ced3463d06de0ca9a15b6153677-Paper-round2.pdf`.

[21] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. *Data mining and knowledge discovery handbook*, pages 667–685, 2010.

[23] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*, pages 145–158. Springer, 2011.

[24] Francisco Charte, Antonio Rivera, María José del Jesus, and Francisco Herrera. On the impact of dataset complexity and sampling strategy in multilabel classifiers performance. In *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings 11*, pages 500–511. Springer, 2016.

[25] Piotr Szymański and Tomasz Kajdanowicz. A network perspective on stratification of multi-label data. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 22–35. PMLR, 2017.

[26] Maximillian Merrillees and Lan Du. Stratified sampling for extreme multi-label data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 334–345. Springer, 2021.

[27] Henri Tiittanen, Liisa Holm, and Petri Törönen. Novel split quality measures for stratified multilabel cross validation with application to large and sparse gene ontology datasets. *arXiv preprint arXiv:2109.01425*, 2021.

[28] Francisco Florez-Revuelta. Evosplit: An evolutionary approach to split a multi-label data set into disjoint subsets. *Applied Sciences*, 11(6):2823, 2021.

[29] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[30] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, 53(11):3735–3745, 2009.

[31] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL `http://manikvarma.org/downloads/XC/XMLRepository.html`.

[32] Richard E Korf. Multi-way number partitioning. In *IJCAI*, volume 9, pages 538–543, 2009.

[33] Brad L Miller, David E Goldberg, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[36] Pavel Iakubovskii. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2019.

[37] G. Rudolph. Convergence of evolutionary algorithms in general search spaces. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 50–54, 1996. doi: 10.1109/ICEC.1996.542332.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims made in the abstract and introduction, namely a stratification algorithm for image segmentation datasets, is supported by the experiments conducted in section 4 and results outlined in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our methods and findings are mentioned in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper offers theorems in the Appendix A. Assumptions are clearly stated in the statements. Theorem statements are mentioned in the main paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed information on model architecture, training procedures, and experimental setup in Section 4. The datasets used in our experiments are publicly available. Additionally, we provide the algorithm and parameters of the proposed methods for easy reproducibility in the Appnedix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We used exclusively publicly available data for training and testing. We also provide a reference implementation allowing for easy reproducibility of our experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed information about dataset used and the experimental setup. Additional information about model hyper-parameters and the optimizer can be found in Section 4. We also provide a reference implementation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report standard deviation across five random seeds for statistical similarity measures and across ten folds for segmentation performance metrics. These error bars reflect variability due to different train/test splits and random initializations, which are central to our main claims. All reported deviations correspond to one standard deviation. The calculation is done using standard statistical functions from Python libraries such as NumPy. This approach captures the expected variance across repeated runs and ensures the reported model performance is not the result of a single experiments. The tables in question are table 2 and table 3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resources are outlined in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:We have followed the NeurIPS Code of Ethics to the best of our ability and knowledge.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The work improves evaluation reliability in image segmentation tasks, especially in critical domains like medical imaging. While primarily methodological, its misuse could enable biased performance reporting if stratification is selectively applied.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The proposed methods and findings pose minimal risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We credit and cite the original owners of assets (PascalVOC, CamVid, EndoVis, LoveDA and Cityscapes) used in this paper. LoveDA and Cityscapes explicitly state in its license that these datasets are freely available for research or academic work. For the PascalVOC, CamVid and EndoVis datasets, no original license is available. The model implementation from `smp` is under MIT License.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code for the proposed algorithm is submitted as part of supplementary material with the necessary documentation. The license for the assets is mentioned wherever applicable. The assets have been anonymized.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Convergence of Stratification Algorithms

This section presents two results on the convergence behavior of stratification methods. The first shows that random splits of large multi-label datasets asymptotically preserve marginal label frequencies in each fold, by the law of large numbers, ensuring representativeness without explicit stratification. The second analyzes WDES, an evolutionary algorithm that minimizes label distribution differences across folds. It proves that the expected quality of the best-found stratification improves at a rate of O(1/MG), converging toward the global optimum as the number of generations and population size grow.

A.1 Asymptotic Representativeness of Random Splits in Multi-Label Datasets

Theorem A.1. *Let $D = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ be a dataset where each $\mathbf{y}_n \in \{0,1\}^L$ is a binary label vector over a fixed set of L labels. Suppose D is randomly partitioned into K disjoint folds $\mathcal{S} = \{S^1, \ldots, S^K\}$ such that each fold satisfies $|S^k| \approx r^k N$, for a fixed target proportion vector $\mathbf{r} \in [0,1]^K$ with $\sum_{k=1}^K r^k = 1$. Then, for any label index $\ell \in \{1, \ldots, L\}$, the empirical label frequency in each fold converges almost surely to the true marginal label frequency as $N \to \infty$. That is, for all $k = 1, \ldots, K$,*

$$\hat{p}_N^k(\ell) \xrightarrow{a.s.} p(\ell),$$

where

$$\hat{p}_N^k(\ell) = \frac{1}{|S^k|} \sum_{n \in S^k} y_{n\ell} \quad \textit{and} \quad p(\ell) = \mathbb{E}[y_{n\ell}] = \mathbb{P}(y_{n\ell} = 1).$$

Proof. Fix a label index $\ell \in \{1, \ldots, L\}$. For each sample n, define the binary indicator:

$$Z_n^{(\ell)} = y_{n\ell} \in \{0,1\}.$$

By assumption, the data points $(\mathbf{x}_n, \mathbf{y}_n)$ are drawn i.i.d., so the sequence $\{Z_n^{(\ell)}\}_{n=1}^N$ is i.i.d. Bernoulli with mean:

$$\mathbb{E}[Z_n^{(\ell)}] = p(\ell).$$

Now consider the random assignment of samples to each fold S^k, where $|S^k| \approx r^k N$. For sufficiently large N, each $|S^k| \to \infty$ and the samples in each fold remain representative of the full dataset.

Let

$$\hat{p}_N^k(\ell) = \frac{1}{|S^k|} \sum_{n \in S^k} Z_n^{(\ell)}.$$

Then, by the Strong Law of Large Numbers (SLLN),

$$\hat{p}_N^k(\ell) \xrightarrow{a.s.} \mathbb{E}[Z_n^{(\ell)}] = p(\ell), \quad \text{as } |S^k| \to \infty.$$

Therefore, for any $k, k' \in \{1, \ldots, K\}$,

$$\left| \hat{p}_N^k(\ell) - \hat{p}_N^{k'}(\ell) \right| \xrightarrow{a.s.} 0,$$

and each fold becomes asymptotically representative of the true label distribution. □

A.2 Optimality and Convergence of WDES

To quantify the effectiveness of WDES, we derive a convergence bound that characterizes how the quality of stratifications improves with the number of generations and population size under standard evolutionary algorithm assumptions.

Proposition A.1 (Convergence Rate of WDES to Empirical Optimum)**.** *Let $\mathcal{F}_{M,G} \subset \mathcal{F}$ denote the set of stratifications explored by the WDES algorithm after G generations with a population size M. Then the expected suboptimality of the final stratification $\mathcal{S}_{\textit{WDES}} \in \mathcal{F}_{M,G}$ satisfies:*

$$\mathbb{E}\left[L(\mathcal{S}_{\textit{WDES}})\right] - L^*_{\mathcal{F}_{M,G}} \leq \mathcal{O}\left(\frac{1}{MG}\right).$$

where $L^*_{\mathcal{F}_{M,G}}$ denotes the optimal Wasserstein score found in the empirical subset.

Proof. WDES is a genetic algorithm that evolves a population of M stratifications over G generations. The algorithm operates using elitist selection, where the best individual is preserved across generations. This ensures that the sequence of best-found scores $\{L^g_{\text{best}}\}_{g=1}^G$ is non-increasing.

Each generation performs M fitness evaluations, totaling MG evaluations across the run. The search space $\mathcal{F}$ is finite because there are finitely many possible assignments of N labeled data points to K folds satisfying approximate size constraints. Hence, WDES induces a finite-state stochastic process over $\mathcal{F}_{M,G} \subset \mathcal{F}$, the set of stratifications visited during execution.

Let $D = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ be a dataset with C classes, and let $\mathcal{F}$ denote the space of feasible stratifications $\mathcal{S} = \{S^1, \ldots, S^K\}$, where each fold S^k satisfies the approximate size constraint $|S^k| \approx r^k N$, for a target proportion vector $\mathbf{r} \in [0,1]^K$, with $\sum_k r^k = 1$.

Define the class distribution in fold k as $P_i^k = \frac{1}{|S^k|} \sum_{n \in S^k} \mathbb{I}[y_n = i]$, and the global class distribution as $P_i = \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y_n = i]$, for $i = 1, \ldots, C$. Let $F_c^k = \sum_{i=1}^c P_i^k$ and $F_c = \sum_{i=1}^c P_i$ be the cumulative class distributions in fold k and the full dataset, respectively.

The Label Wasserstein Distance of stratification $\mathcal{S}$ is defined as:

$$L(\mathcal{S}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} \left| F_c^k - F_c \right|.$$

Crucially, the mutation operator used in WDES is assumed to be ergodic, i.e., it has a positive probability of reaching any feasible stratification from any other. This property, together with elitist selection, guarantees that the algorithm converges in probability to the global optimum over $\mathcal{F}$, as shown in [37]. However, in finite time, WDES only explores $\mathcal{F}_{M,G}$, and its best possible result is $L^*_{\mathcal{F}_{M,G}}$. The expected gap between the final stratification's score and this empirical optimum decreases with the number of independent evaluations MG. This follows from classical convergence results for evolutionary algorithms under elitist, ergodic dynamics, where the convergence rate to the best solution seen is:

$$\mathbb{E}\left[L(\mathcal{S}_{\text{WDES}})\right] - L^*_{\mathcal{F}_{M,G}} \leq \mathcal{O}\left(\frac{1}{MG}\right).$$

This bound reflects the diminishing returns of increasing population size and generations, but confirms that more search budget leads to better empirical solutions. □

Remark. *The theorem guarantees convergence to the best stratification within the explored subset $\mathcal{F}_{M,G}$, not necessarily the global optimum $L^* = \min_{\mathcal{S} \in \mathcal{F}} L(\mathcal{S})$. Nonetheless, due to the ergodic nature of the mutation operator, the probability of reaching any region of the space is non-zero. Thus, $L^*_{\mathcal{F}_{M,G}} \to L^*$ as $G \to \infty$, and the algorithm converges to the global optimum in probability.*

B Iterative Pixel Stratification Algorithm

Following the notation introduced in Section 3, the algorithm for IPS is described in Algorithm 1.

Algorithm 1 Iterative Pixel Stratification

Input: A dataset D of C classes, N samples that have a total of P pixels. The dataset is to be divided into K folds in the proportion $\mathbf{r} = r_1, \ldots, r_k$.

Output: Disjoint subsets $S_1, \ldots, S_k$

```
# Calculate desired number of samples per fold
{N^k}_1^K ← N · r
for c ← 1 to C do
    # Calculate number of pixels present per class
    P_c ← {P : c ∈ C}
    # Calculate desired number of pixels per class per fold
    {P_c^k}_1^K ← P_c · r
end for
while |D| > 0 do
    # Find rarest class
    l ← argmin_c P_c
    for (x_l, y_l) ∈ D_l do
        # Find folds with largest desire for class l
        M ← argmax_{k∈K} P_l^k
        if |M| = 1 then
            m ← M
        else
            M' ← argmin_{k∈M} N^k
            if |M'| = 1 then
                m ← M'
            else
                m ← Random Element of (M')
            end if
        end if
        # Assign sample to subset and remove from dataset
        S^m ← S^m ∪ (x_l, y_l)
        D ← D \ {(x_l, y_l)}
        N^m ← N^m − 1
        for c ∈ C do
            P_c^m = P_c^m − P_c(x_l, y_l)
        end for
    end for
end while
```

C Wasserstein-Driven Evolutionary Stratification Algorithm

Following the notation introduced in Section 3, the algorithm for WDES is described in Algorithm 2.

Algorithm 2 Wasserstein-Driven Evolutionary Stratification (WDES)

Input: A dataset D of C classes and N samples, to be divided into K folds according to proportions $\mathbf{r} = (r_1, \ldots, r_K)$; population size M; number of generations G; crossover rate p_c; mutation rate p_m.

Output: Disjoint subsets $S_1, \ldots, S_K$ corresponding to the fittest individual.

```
# Initialize population of candidate stratifications
for i ← 1 to M do
    Generate random fold assignments according to r
    Store as individual I_i
end for
for g ← 1 to G do
    # Evaluate fitness of each individual
    for i ← 1 to M do
        Evaluate fitness using Label Wasserstein Distance (Eq. 3) for fold assignments in I_i
    end for
    # Select parents via tournament selection
    Select pairs of individuals based on lowest LWD values
    # Apply crossover and correction
    for p ← 1 to number of parent pairs do
        if random() < p_c then
            Exchange random segments of fold assignments between parents
            Apply correction to preserve fold proportions r
        end if
    end for
    # Apply mutation
    for i ← 1 to M do
        if random() < p_m then
            Randomly swap fold assignments of two samples in I_i
        end if
    end for
end for
# Select best-performing individual as final stratification
I* ← argmin_{I_i} LWD(I_i)
Return S_1, ..., S_K as the folds encoded in I*
```

D Genetic Algorithm parameters

We empirically selected the hyper-parameters for the genetic algorithm by evaluating performance across multiple datasets. For each parameter, we increased its value until no further improvements were observed in the stratification quality. The final values are shown in Table 5,

Table 5: WDES Parameters

Parameter	Value
Number of Generations	50
Number of Individuals	100
Gene Mating Probability	0.5
Individual Mutation Probability	0.2
Selection Tournament size	3

E Time and Space Complexity of WDES

The computational complexity of the Wasserstein Distance-based Evolutionary Stratifier (WDES) can be analyzed in terms of the population size M, number of generations G, number of samples N, number of classes C, and number of folds K. Each individual in the population represents an assignment of all N samples into K folds, and the fitness evaluation involves computing the Wasserstein distance across C-dimensional class histograms.

The overall time complexity is therefore $\mathcal{O}(M \cdot G \cdot K \cdot C)$, with an additional linear component in N arising from crossover and mutation operations. The space complexity is $\mathcal{O}(M \cdot N)$, as the algorithm maintains a population of M individuals, each encoding fold assignments for all samples.

In practice, the runtime of WDES is further influenced by the genetic algorithm parameters, including mutation and crossover probabilities, which determine how much of the population is updated between generations. These hyperparameters affect convergence speed and computational cost but do not alter the asymptotic complexity. An empirical assessment of runtime performance is provided in the following section.

F Runtime Analysis

We evaluate the runtime performance of our stratification methods on an Apple MacBook Pro with an M3 Pro processor and 36 GB of memory. Figure 4 presents the runtime in seconds (with a logarithmic scale on the y-axis) for Random splitting, IPS, and WDES across all datasets.

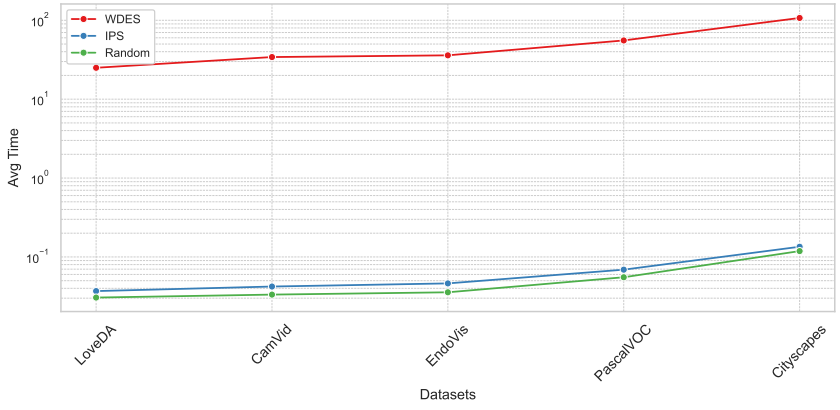

Figure 4: Runtime comparison of Random splitting, IPS, and WDES across all datasets.

Table 6: Comparison of stratification methods on similarity-based evaluation metrics. Results are averaged across datasets for Random splitting, IPS, CS-IPS, and WDES. While CS-IPS performs slightly better than IPS in some cases, WDES consistently achieves the best scores across all metrics, indicating stronger distributional alignment.

Dataset	Method	SD		PLD $(\times 10^{-5})$		LWD $(\times 10)$	
		Mean	Std	Mean	Std	Mean	Std
PascalVOC	Random	**0.42**	**0.00**	955	137	75.0	2.62
	CS-IPS	0.96	0.00	525	6.33	42.5	0.32
	IPS	10.28	2.46	733	244	53.3	1.20
	WDES	**0.42**	**0.00**	**456**	**77.9**	**51.4**	**1.63**
Camvid	Random	**0.00**	**0.00**	63.4	3.73	183	5.08
	CS-IPS	0.60	0.00	58.0	0.39	175	1.48
	IPS	6.68	1.18	76.6	5.30	209	8.44
	WDES	**0.00**	**0.00**	**36.7**	**4.47**	**138**	**7.84**
EndoVis	Random	**0.50**	**0.00**	609	81.4	661	35.5
	CS-IPS	0.80	0.00	485	8.00	549	3.34
	IPS	11.42	1.05	769	12.2	764	75.2
	WDES	**0.50**	**0.00**	**208**	**23.8**	**399**	**10.4**
LoveDA	Random	**0.32**	**0.00**	1110	109	1080	88.4
	CS-IPS	0.64	0.00	935	11.9	892	7.58
	IPS	8.06	1.15	1090	207	1100	87.4
	WDES	**0.32**	**0.00**	**377**	**31.4**	**635**	**16.0**
Cityscapes	Random	**0.50**	**0.00**	126	15.4	304	16.5
	CS-IPS	1.90	0.00	130	4.48	300	4.73
	IPS	11.54	2.69	148	17.4	324	22.2
	WDES	**0.50**	**0.00**	**67**	**3.44**	**217**	**5.17**
Ranking	Random	**1**		3.0		3.2	
	CS-IPS	2		2.2		2.0	
	IPS	3		3.8		3.8	
	WDES	**1**		**1**		**1**	

G Class-Sensitive Iterative Pixel Stratification (*CS-IPS*)

The Class-Sensitive Iterative Pixel Stratification (CS-IPS) algorithm introduces an alternative iterative approach for constructing balanced folds based on pixel-level class distributions. The method begins by ranking all classes in the dataset according to their rarity, determined by the total number of pixels belonging to each class. Starting from the rarest class, all samples containing pixels of that class are identified and sorted in descending order based on the number of pixels associated with the current class. The sorted samples are then distributed across folds in a zig-zag manner to ensure a balanced representation of the class across partitions. This procedure is repeated iteratively for each subsequent class until all samples have been assigned. The resulting folds aim to maintain pixel-level balance while accounting for class frequency in a straightforward, iterative manner.

We report the results of CS-IPS alongside those obtained using random splitting, IPS, and WDES in Table 6. While CS-IPS achieves slightly improved performance compared to IPS in several cases, WDES consistently attains the highest overall performance across datasets. These results reinforce our motivation for adopting a more flexible, globally optimized approach such as the genetic algorithm employed in WDES.

The corresponding segmentation results, summarized in Table 7, exhibit a similar trend. CS-IPS provides measurable improvements over IPS, indicating the benefits of incorporating class sensitivity in the iterative process. However, WDES continues to outperform all other methods, underscoring the advantages of optimization-driven stratification over purely heuristic approaches.

Table 7: Average segmentation performance of UNet under 10-fold cross-validation using different stratification methods. Reported values correspond to Accuracy, F1 Score, and IoU. CS-IPS shows improved stability compared to IPS but remains below WDES, which continues to yield the most consistent and reliable fold performance.

Dataset	Method	Accuracy		F1		IoU	
		Mean	Std ($\times 10^{-3}$)	Mean	Std ($\times 10^{-3}$)	Mean	Std ($\times 10^{-3}$)
PascalVOC	Random	0.76	20.6	0.58	32.5	0.44	34.3
	CS-IPS	0.71	12.2	0.45	26.2	0.32	25.3
	IPS	0.76	14.5	0.58	30.3	0.44	31.6
	WDES	0.75	**11.0**	0.57	**24.0**	0.43	**24.2**
Camvid	Random	0.94	0.68	0.91	1.16	0.89	1.17
	CS-IPS	0.94	0.70	0.91	1.32	0.89	1.24
	IPS	0.94	0.74	0.91	1.35	0.89	1.27
	WDES	0.94	**0.67**	0.91	**1.09**	0.89	**1.06**
EndoVis	Random	0.94	19.4	0.86	27.0	0.80	28.1
	CS-IPS	0.92	13.6	0.86	21.2	0.78	21.4
	IPS	0.94	14.1	0.87	29.3	0.79	30.7
	WDES	0.94	**13.0**	0.85	**17.9**	0.79	**18.9**
LoveDA	Random	0.88	**6.63**	0.80	**6.39**	0.69	**8.38**
	CS-IPS	0.88	7.30	0.80	8.19	0.69	10.3
	IPS	0.88	7.39	0.80	9.29	0.69	10.7
	WDES	0.88	7.49	0.80	9.86	0.69	11.0
Cityscapes	Random	0.79	11.2	0.61	**9.12**	0.50	**8.64**
	CS-IPS	0.79	12.2	0.61	21.5	0.51	18.2
	IPS	0.79	**10.3**	0.61	14.9	0.50	15.2
	WDES	0.79	11.2	0.61	14.4	0.50	12.8
Ranking	Random		2.6		2.2		2.2
	CS-IPS		2.4		2.6		2.6
	IPS		2.6		3.2		3.6
	WDES		**1.8**		**1.8**		**1.8**

H Additional Evaluation Metrics

To ensure that the evaluation of stratification quality is not biased toward a specific similarity metric, we further assess all stratification methods using additional distributional measures: the Linear-kernel Maximum Mean Discrepancy (L-MMD) and the Linear Energy Distance (L-ED). These metrics provide complementary perspectives on distributional similarity between folds and the overall dataset, capturing higher-order differences beyond those measured by the Label Wasserstein Distance (LWD).

Table 8 reports results for all methods—random splitting, IPS, CS-IPS, and WDES—across these alternative metrics. Consistent with the findings based on LWD, WDES achieves the lowest average discrepancy across datasets, demonstrating that its advantages generalize across diverse statistical criteria. IPS performs competitively on the PascalVOC dataset, which we attribute to its higher proportion of single-class samples. Overall, these results confirm that the performance gains of WDES are robust to the choice of similarity metric.

Pixel Label Pair Distribution (*PLPD*) In addition to the aforementioned distributional measures, we incorporate a class co-occurrence metric inspired by the Label Pair Distribution (LPD) proposed in literature [25]. This measure captures second-order relationships between classes that tend to appear together within the same sample. We adapt this concept to the pixel level, computing the Pixel Label Pair Distribution (PLPD) to quantify the co-occurrence consistency between folds and the global dataset. The results, presented in Table 8, extend our analysis by evaluating how well each method preserves inter-class relationships. The previously introduced CS-IPS method is also included for completeness.

Table 8: Evaluation of stratification quality using alternative similarity measures: Pixel Label Pair Distribution (PLPD), Linear-kernel Maximum Mean Discrepancy (L-MMD), and Energy Distance (L-ED). Results are averaged across datasets for Random splitting, IPS, CS-IPS, and WDES. WDES consistently outperforms the other methods across all three metrics, demonstrating that its advantages generalize beyond Wasserstein-based evaluation.

Dataset	Method	PLPD	L-MMD $(\times 10^2)$	L-ED
PascalVOC	Random	0.67	5.67	3.64
	CS-IPS	0.69	2.68	2.91
	IPS	**0.66**	**1.84**	2.56
	WDES	0.76	2.94	**2.49**
Camvid	Random	4.70	12.2	8.93
	CS-IPS	5.97	11.1	8.28
	IPS	4.62	14.7	9.91
	WDES	**4.58**	**7.16**	**6.98**
EndoVis	Random	0.36	46.8	14.7
	CS-IPS	0.29	35.7	11.7
	IPS	0.27	55.2	16.4
	WDES	**0.25**	**17.9**	**9.53**
LoveDA	Random	$1.35_{\times 10^{-5}}$	78.6	20.9
	CS-IPS	$1.22_{\times 10^{-5}}$	62.7	17.2
	IPS	$1.21_{\times 10^{-5}}$	69.9	21.3
	WDES	$\mathbf{1.20}_{\times \mathbf{10}^{-5}}$	**27.7**	**12.8**
Cityscapes	Random	3.73	20.7	7.70
	CS-IPS	3.75	21.2	7.44
	IPS	3.85	23.3	8.24
	WDES	**3.68**	**11.8**	**5.79**
Ranking	Random	3	3.2	3.2
	CS-IPS	3.2	2.2	2.2
	IPS	2.2	3.2	3.6
	WDES	**1.6**	**1.4**	**1.0**

These additional evaluations further corroborate the overall trends observed in the main results: while iterative approaches such as IPS and CS-IPS produce reasonable fold-level balance, WDES consistently achieves superior alignment with global label distributions across all tested metrics.

I Dataset details

I.1 Cityscapes

Cityscapes is a large benchmark dataset for training and testing pixel-level and instance-level semantic labeling. It contains diverse stereo video sequences from street scenes in 50 cities. Of these frames, 5,000 images have high-quality pixel annotations across 30 visual classes, grouped into eight categories: flat, construction, nature, vehicle, sky, object, human, and void. We use only the left-camera images (the annotated view) and run our experiments on the training set, since it's the largest split.

I.2 CamVid

The Cambridge-driving Labeled Video Database (CamVid) captures footage from the perspective of a driving automobile, unlike most videos that are filmed with fixed-position CCTV-style cameras. From this footage, 701 frames were sampled at 1 Hz and manually labeled with 32 semantic classes. Similar to CityScapes, we only use the largest subset, the training set.

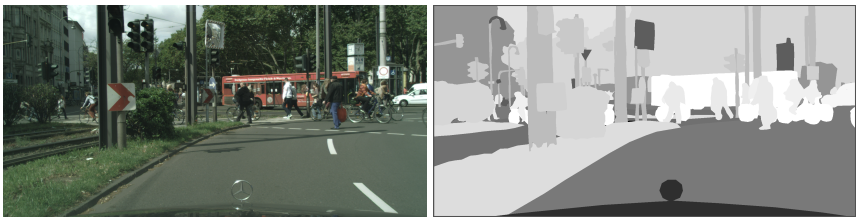

Figure 5: Example Cityscapes data: the original image (left) and its enhanced annotation mask (right), for better visualization.

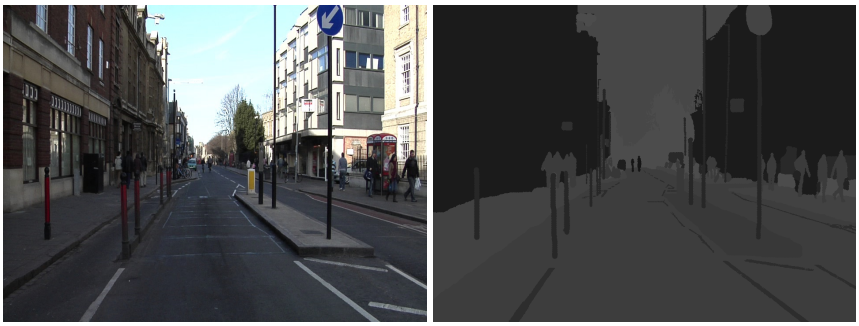

Figure 6: Example CamVid data: the original image (left) and its enhanced annotation mask (right).

I.3 Pascal VOC 2012

The Pascal VOC 2012 serves as a cornerstone resource for training and comparing semantic segmentation models. It comprises 20 classes, including entities like people, animals, vehicles, and indoor objects. The dataset comprises 1,464 training images, 1,449 validation images, and a private testing set. Each image in this dataset is annotated with pixel-level segmentation, bounding box, and object class information. For our experiments, we exclusively utilized the training set.

Figure 7: Example Pascal VOC 2012 data: the original image (left) and its enhanced annotation mask (right).

I.4 EndoVis2018

The EndoVis2018 dataset, used in the Robotic Scene Segmentation Challenge of MICCAI 2018, contains 16 robotic nephrectomy procedures recorded using da Vinci Xi systems in porcine labs. Each procedure comprises 149 training frames and 250 testing frames, each with a resolution of 1280×1024. The dataset includes images from both the left and right eye cameras, as well as the stereo camera calibration parameters. However, labels are only available for the left eye camera, with 12 categories. In this case, we utilized only the training set.

I.5 LoveDA

The LoveDA dataset, a comprehensive collection of remote sensing images, is designed for semantic segmentation. It comprises 5,987 high-resolution images captured across three Chinese cities,

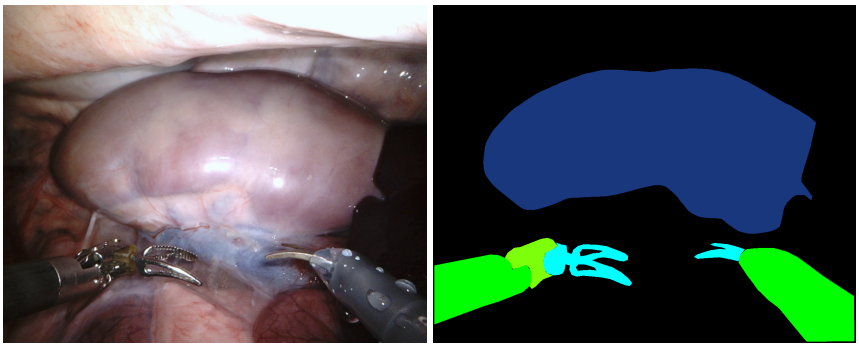

Figure 8: Example EndoVis2018 data: the original image (left) and its annotation mask (right).

capturing a diverse range of urban and rural scenes. These images are meticulously annotated with 166,768 object annotations, categorized into seven distinct land-cover classes: background, buildings, roads, water bodies, barren land, forests, and agricultural areas. Again, only the training set is utilized.

Figure 9: Example LoveDA data: the original image (left) and its enhanced annotation mask (right).

J Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR) of the FAU Erlangen-Nürnberg. The hardware is funded by the Deutsche Forschungsgemeinschaft (DFG).