Appendix A: Dataset Exclusion Criteria

While the two reannotated datasets introduced by Ma *et al.* [21] initially provided a valuable resource for determining convergence thresholds, we encountered several issues that prevented accurate threshold determination:

1. **Different Annotation Guidelines:** The datasets did not adhere to the same guidelines. Since they were annotated by different groups with varying annotation pipelines and guidelines, the annotation variations cannot be attributed to regular issues shown in Figure 2. These are not ambiguities within a single guideline but rather differences between distinct guidelines, resulting in label conventions that deviate significantly from the guideline.

2. **Sampling Bias:** The reannotation process exhibits a sampling bias. Images were selected for reannotation based on the presence of at least one of the five chosen classes. This selection process focused on false positives while potentially overlooking false negatives, thereby skewing the dataset.

3. **Annotation Inconsistencies:** There were inconsistencies in annotation formatting, with some annotations being untraceable to their corresponding images and vice versa. This suggests that some annotation files were incomplete.

4. **Suspicious IoU Matches:** Anomalously high instances of perfect IoU (Intersection over Union) matches (1.0) were noted, indicating possible annotation duplication from the original datasets, although this was not explicitly confirmed in their documentation. LVIS, TexBiG, and VinDr-CXR did not contain a single instance with a 1.0 IoU overlap.

5. **Limited Class Coverage:** Only five classes were selected for reannotation, reducing the Open Images dataset to approximately 5,000 images due to resource constraints. Extrapolating the convergence threshold from these five classes to the entire dataset decreases the validity of the estimated convergence threshold.

Due to these points, the reannotated datasets present limited validity and generalizability. Consequently, we decided not to determine label convergence using these reannotated versions, as we do not see results on these datasets as reflective of the remaining commonly used COCO dataset. However, we still use the data to fit the linear regression, as they reflect real annotation variations, which we prefer over synthetic data.

Appendix B: Recap of Krippendorff's Alpha for Object Detection

To evaluate annotation consistency, we use the method introduced by Tschirschwitz *et al.* [38], which adapts Krippendorff's Alpha (K-α) for object detection. This method calculates a single α value to measure inter-annotator agreement, where $\alpha = 1$ indicates perfect agreement, $\alpha = 0$ indicates no agreement, and $\alpha < 0$ indicates disagreement. The general form of K-α is $\alpha = 1 - \frac{D_o}{D_e}$, where D_o is the observed disagreement and D_e is the expected disagreement.

Calculation Procedure

Using our prior definition of annotations from Section 1 where a single annotation is described as $\tilde{y}_{ij}^r$ which refers to annotation j for image i annotated by annotator r, the following steps are executed for a single image i:

1. **Localization Overlap Calculation:** The intersection over union (IoU) is calculated between different annotators r for each of their respective instances. For example, take annotator A and annotator B.

$$IoU(\tilde{y}_{ij}^A, \tilde{y}_{ij}^B) = \frac{|\tilde{y}_{ij}^A \cap \tilde{y}_{ij}^B|}{|\tilde{y}_{ij}^A \cup \tilde{y}_{ij}^B|} \tag{2}$$

2. **Cost Matrix and Matching:** A cost matrix is created using the function:

$$C(j,k) = 1 - IoU(\tilde{y}_{ij}^A, \tilde{y}_{ik}^B) \tag{3}$$

   Assume that annotator A has M_A annotations and annotator B has M_B annotations for image i. The sets are matched using the Hungarian algorithm, ensuring $M_A = M_B$ by padding the smaller set with $\varnothing$. For multiple annotators ($R > 2$), a greedy matching is algorithm is applied between the matched sets.

3. **Reliability Data and Coincidence Matrix:** After matching, reliability data is organized into a coincidence matrix with values o_{ck} representing the number of c-k pairs (referring here to a pair of categories assigned to the same unit by different annotators) for each instance (unit) u, calculated as:

$$o_{ck} = \sum_u \frac{\text{Number of c-k pairs in unit u}}{m_u - 1} \tag{4}$$

   where m_u is the number of annotators (observers) for unit u, so how many annotators found the same instance u. From this, we calculate:

$$n_c = \sum_k o_{ck} \quad \text{and} \quad n = \sum_c n_c \tag{5}$$

   Here, n_c represents the total number of times category c was assigned across all units, and n is the total number of paired observations across all categories.

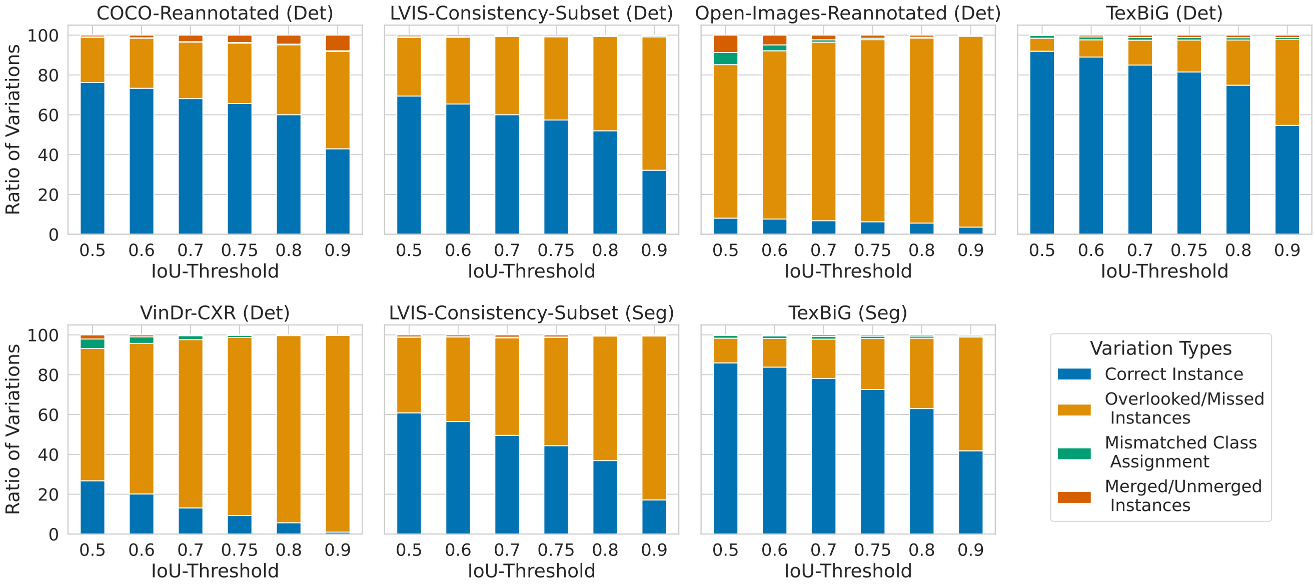

Figure 6. Variation distribution across the remaining datasets, with (Det) referring to object detection and (Seg) referring to instance segmentation, indicating similar trends to those observed with TexBiG and LVIS. However, these two datasets are of relatively high agreement with good annotation quality.

4. **Krippendorff's Alpha Calculation:** Finally, α for nominal data is calculated using:

$$\alpha = 1 - \frac{D_o}{D_e} = \frac{(n-1)\sum_c o_{cc} - \sum_c n_c(n_c - 1)}{n(n-1) - \sum_c n_c(n_c-1)} \tag{6}$$

Further information about the method can be found in the paper [39].

Interpretation of Alpha Values

- $\alpha \geq 0.8$ signifies reliable and strong agreement among raters.
- $\alpha \geq 0.667$ is considered acceptable with moderate agreement.
- $\alpha = 0$ indicates agreement no better than chance, suggesting random assignment of classes.
- $\alpha < 0$ denotes systematic disagreement, which could indicate unclear guidelines, insufficient rater expertise, or particularly challenging images.

To ensure the accuracy of this method, the method discourages missing entries by replacing them with a filler class, leading to worse agreement scores if an annotator misses an entry that others found.

Appendix C: Additional Material - Annotation Variation Type Analysis

For counting the variations, we employ an algorithm designed to match as many instances as possible. The algorithm requires three elements for each image: 1) the annotations, 2) an IoU threshold, and 3) a list of annotators assigned to this image. For each possible pair of annotators, their respective instance IoU is calculated. Using this localization information:

1. **Matching of Correct Instances:** Instances of the same class are matched starting with the highest overlapping pair of instances until the last pair with an IoU value greater than or equal to the IoU threshold. These instances are then excluded from further matching.

2. **Matching of Merged/Unmerged Instances with Correct Classes:** In the next step, all remaining instances from each annotator are merged within their own class. These merged instances are then included in the IoU evaluation, and the same matching procedure is executed again, excluding possible matches.

3. **Matching of Wrong-Class Instances:** Instances with correct localization but mismatching classes are matched next, following the same procedure, this excludes the previously merged instances.

4. **Matching of Merged/Unmerged Instances with Incorrect Classes:** Similar to step 2, merged instances

are created within the annotations of a single annotator but are now allowed to match with instances from other classes from the other annotator.

5. **Missing/Additional Instances:** All remaining instances are counted as missing or additional, as they did not find any match.

With this hierarchical procedure, we aim to match as many instances as possible, essentially adopting a lenient approach toward annotation mistakes. This means that while an instance with a better overlap might be available, the chosen match will correspond to the class of the annotation. This approach maximizes agreement wherever possible. Therefore, matches with higher IoU are generally preferred, but matches with fitting classes take precedence if they exceed the IoU threshold.

In Figure 6, we present the variation distribution across the different analyzed datasets. Figure 7 visualizes the boundary qualities observed with an IoU threshold of 0.5. The remaining three images illustrate various types of variations.

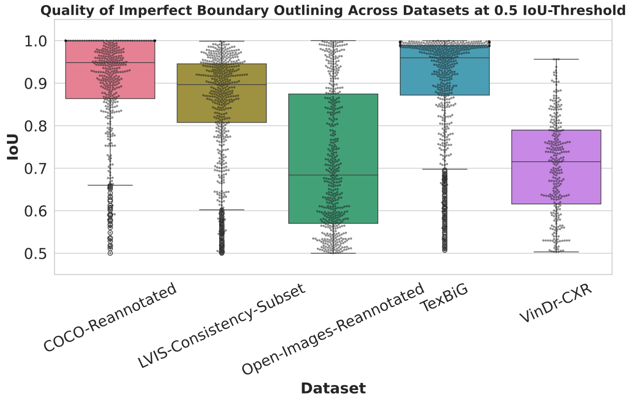

Figure 7. Boundary quality, illustrating how good the localization quality within correct classes is. COCO-Reannotated shows a very high number of 1.0 IoU overlaps, suggesting possible duplication from the original dataset to the reannotated version.

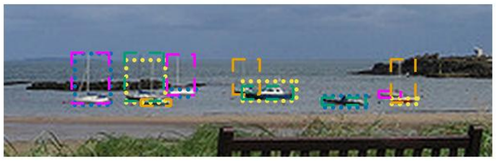

Figure 8. The dotted line represents annotator A while the dashed line represents annotator B. We can see that the boats are very hard to recognize when not zoomed into the image (full image 9. We consider this an annotation variation caused by image quality or at least perceived image quality, as this might also be related to the available tooling for the annotation process.

Figure 9. This image shows a full image without any annotation, and Figure 8 a zoomed in version.

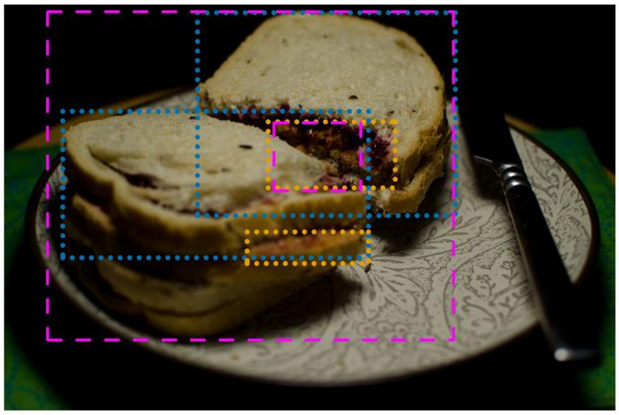

Figure 10. The dotted line represents annotator A while the dashed line represents annotator B. This image again shows a case of a merging issue, where both annotators made reasonable assumptions about the labeling convention, however the guideline seems to be not specific enough. The magenta instance in the center and the two orange instances are parts of the class peanut butter. Here the interpretation seems very difficult, almost like an occlusion case. One annotator opted for additional peanut butter at the bottom sandwich, while the other annotator did not find any peanut butter there. We also attribute this issue to image quality.