

PLATO++: Pose-Conditioned Part-Aware Object Generation via Residual Structure Learning

Supplementary Material

Anonymous CVPR submission

Paper ID 15

A. Our Part-Aware Segmentation Metrics for Object Generations

The typical image *FID* is an aggregate statistical metric and does not provide a measure instance level object quality and in particular, it does not measure part adherence. To overcome these shortcomings, **we introduce two novel part-aware segmentation-based metrics**. To compute these metrics, we pass the generated image to a SOTA object part segmentation network [1] to obtain part-level bounding boxes and associated part labels.

Part-Layout-F1 is defined as a part class presence metric and is computed similar to the standard F-1 score, i.e. $PL-F1 = 2|P'_c \cap P_c|/(|P'_c| + |P_c|)$ where P_c is the list of parts present in the generated layout, P'_c is list of parts in object segmentation output and c is the object class.

The second novel metric is **Part Layout-IOU (PL-IOU)**, an overlap metric. To compute this metric, we determine the per-part IOU between segmentation output part boxes and their counterparts from the generated layout. A high value of this metric implies that the input conditioning was of sufficient specificity and quality to provide the desired controllability at the part level and retrain the structure of the conditioning layout in the generated image.

Since our metrics are at an instance level, they can be used to rank generations by part-presence based quality. See examples in Fig. 1

B. Ablation Study: Residual Adjacency Formulation

We perform an ablation study to evaluate the impact of different adjacency modeling strategies in the layout generator. Specifically, we compare three variants:

- **Fixed Adjacency:** Uses a class-specific canonical adjacency matrix \mathbb{A}_{fixed} with no learnable adaptation.
- **Full Adjacency Prediction:** Predicts the entire adjacency matrix $\hat{\mathbb{A}}$ directly from the latent representation.
- **Residual Adjacency (Ours):** Predicts a residual correction $\Delta\mathbb{A}$ over a canonical base structure, such that $\hat{\mathbb{A}} = \mathbb{A}_{fixed} + \Delta\mathbb{A}$.

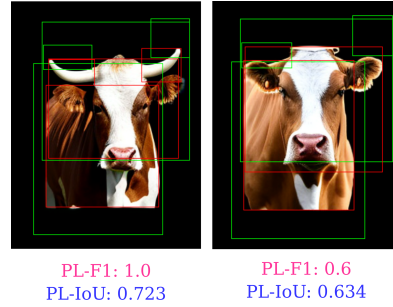


Figure 1. **Measuring part-level fidelity in object generations with our metrics:** The figure shows two images where the horns are missing in the second image. The green bounding boxes represent the ground truth and the red bounding boxes are derived from OLAF’s [1] segmentation result. Our PartGen-F1 score is effectively able to capture the loss of these parts.

rection $\Delta\mathbb{A}$ over a canonical base structure, such that $\hat{\mathbb{A}} = \mathbb{A}_{fixed} + \Delta\mathbb{A}$.

Results. As shown in Table 1, the residual formulation significantly outperforms both fixed and fully learned adjacency strategies. The fixed adjacency baseline achieves reasonable performance but is limited by its inability to adapt to pose-dependent structural variations, leading to suboptimal layout reconstruction. In contrast, directly predicting the full adjacency matrix results in slightly worse performance, likely due to optimization instability and the lack of strong structural priors.

Our residual adjacency formulation achieves the highest mIoU, demonstrating that combining a canonical structural prior with learnable residual corrections provides an effective balance between stability and flexibility. This enables the model to capture non-canonical part interactions while preserving anatomically plausible connectivity.

Adjacency Strategy	mIoU \uparrow
Fixed Adjacency (\mathbb{A}_{fixed})	0.47
Full Adjacency Prediction ($\hat{\mathbb{A}}$)	0.46
Residual Adjacency (Ours)	0.57

Table 1. **Ablation on adjacency modeling strategies.** Residual adjacency significantly improves layout reconstruction performance compared to both fixed and fully learned adjacency formulations.

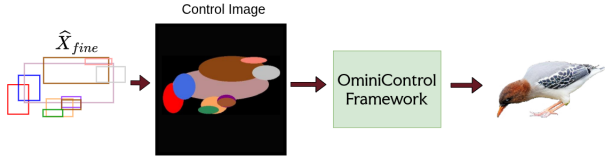


Figure 2. **Layout-to-image control pipeline.** The predicted layout \hat{X}_{fine} is converted into a control image by mapping each part bounding box to a filled geometric primitive (ellipse) with a unique semantic encoding. This control representation is used to guide a layout-conditioned diffusion model, ensuring that the generated image adheres to the specified part layout and pose.

C. Layout-to-Image Generation Details

We provide additional details on the transformation from structured layouts to the control representation used for image synthesis. An overview of this process is shown in Fig. 2. Given the predicted part-level layout \hat{X}_{fine} from the layout generator, we construct a dense control image that encodes the spatial configuration of object parts. Each part bounding box is deterministically mapped to a filled geometric primitive (ellipse) inscribed within the bounding box. This transformation preserves the spatial extent and relative positioning of parts while producing a smooth and compact representation suitable for conditioning diffusion models.

Each semantic part is assigned a distinct color or channel encoding, allowing the control image to retain part identity information. This ensures that different parts (e.g., head, torso, limbs) remain distinguishable during generation.

The resulting control image serves as a structural prior and is provided as input to a layout-conditioned diffusion transformer based on the OminiControl framework. In addition, a textual prompt is constructed from the object category and part list, providing semantic guidance.

During generation, the control image enforces adherence to the predicted layout, while the diffusion model synthesizes fine-grained appearance details such as texture, lighting, and shape. This separation of structure (from the layout) and appearance (from the diffusion model) enables robust generation of articulated objects that respect both spatial constraints and semantic descriptions.

This pipeline allows PLATO++ to translate structured, part-aware layouts into high-quality images while preserving pose, part composition, and spatial relationships.

D. Ground-Truth Adjacency Construction

In this section, we describe the procedure used to construct ground-truth adjacency matrices for supervising the learnable adjacency module.

Given an object instance represented by a set of part bounding boxes $\{b_i\}_{i=1}^p$, we construct an adjacency matrix $\mathbb{A} \in \{0, 1\}^{p \times p}$ that encodes pairwise structural relationships between parts. Since explicit annotations for part connectivity are not available in the dataset, we derive adjacency using a geometric heuristic based on spatial proximity.

Bounding Box Overlap Heuristic. Two parts i and j are considered connected if their corresponding bounding boxes exhibit sufficient spatial overlap. Specifically, we compute the Intersection-over-Union (IoU) between bounding boxes b_i and b_j , and define:

$$\mathbb{A}_{ij} = \begin{cases} 1 & \text{if } \text{IoU}(b_i, b_j) > \delta, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where δ is a predefined threshold.

This heuristic captures the intuition that physically connected or interacting parts tend to occupy overlapping or adjacent spatial regions in the image. For example, the head and torso or torso and limbs typically share boundary regions, resulting in non-zero overlap.

This dynamic supervision plays a crucial role in enabling the residual adjacency formulation to capture non-canonical configurations observed in articulated objects.

E. Illustrating PLATO’s Generalizability

Sample generations are shown for each category. Last row contains failure generations.

E.1. Object generations by Category

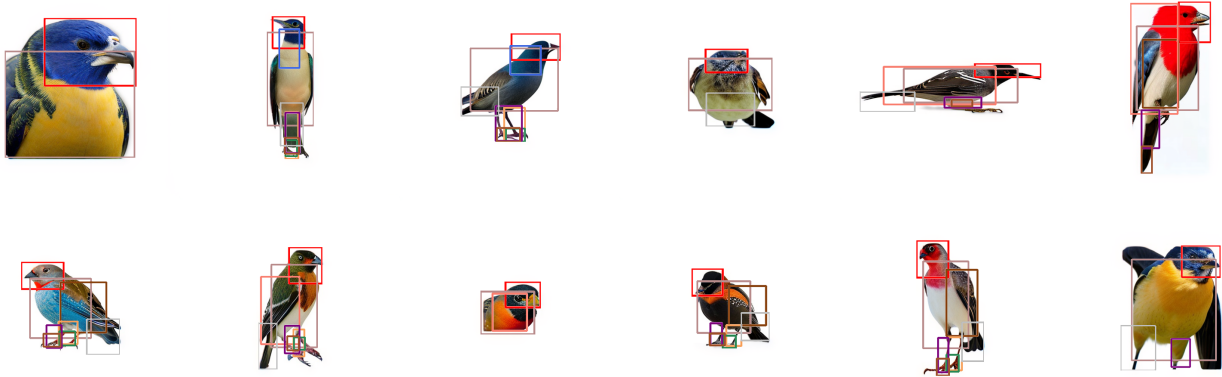


Figure 3. Bird Generations

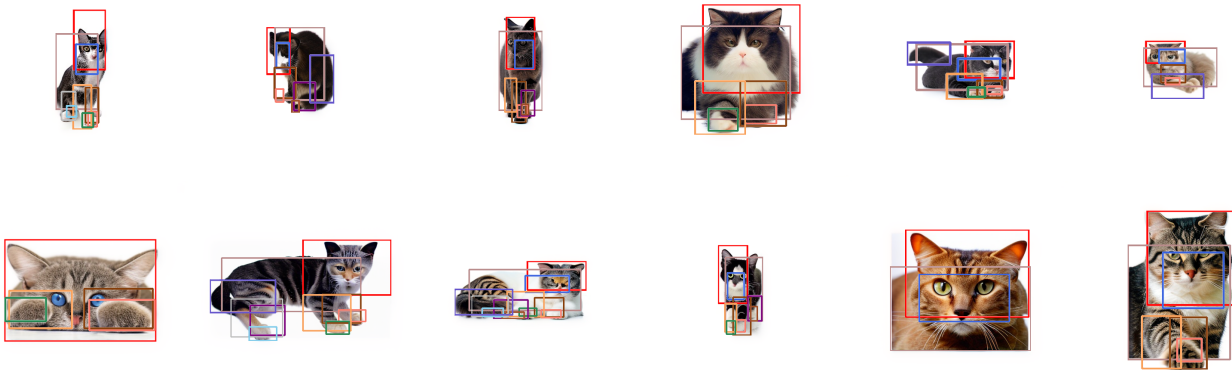


Figure 4. Cat Generations

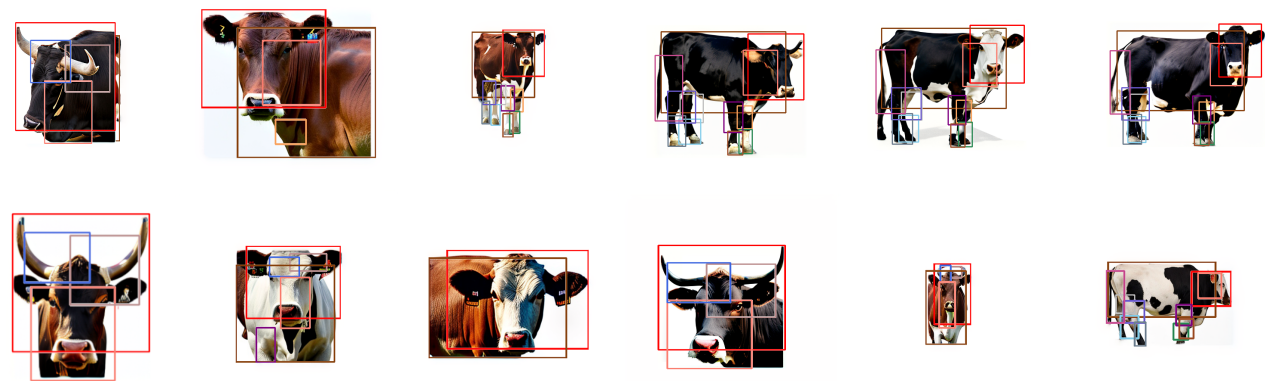


Figure 5. Cow Generations

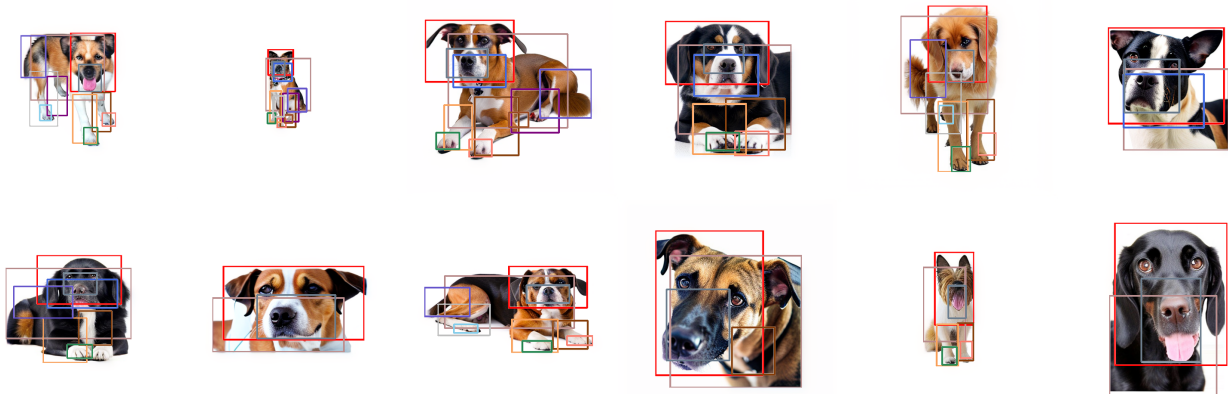


Figure 6. Dog Generations

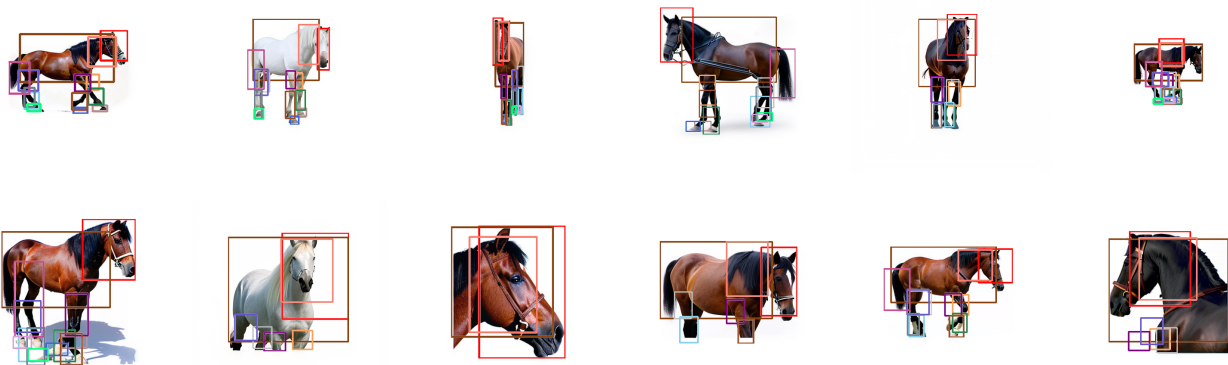


Figure 7. Horse Generations

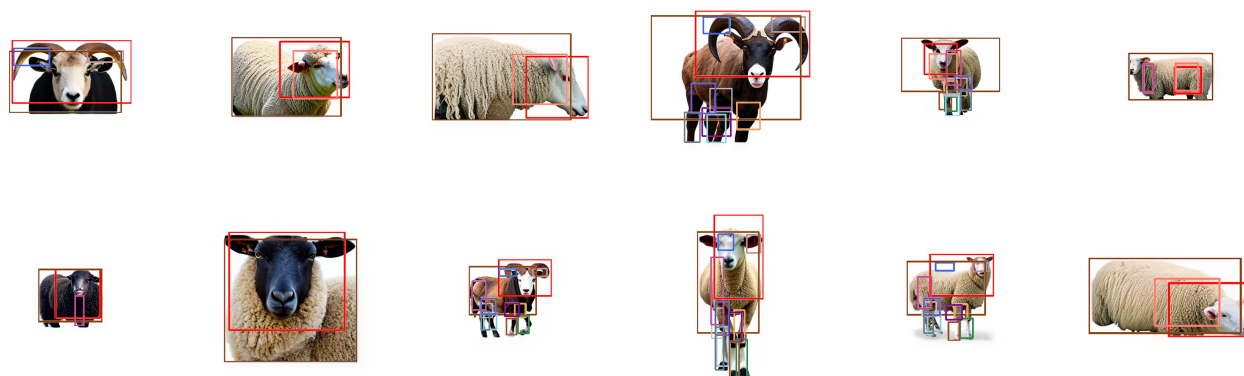


Figure 8. sheep Generations

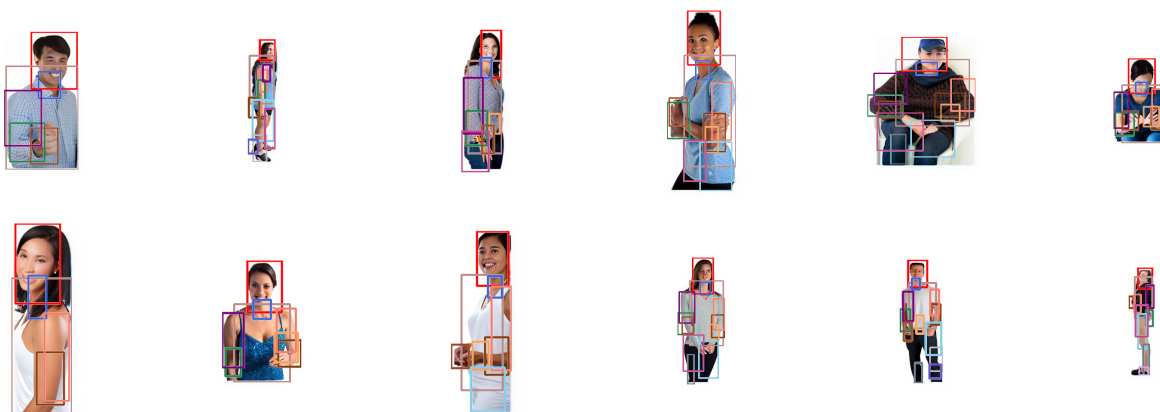


Figure 9. Person Generations



Figure 10. Aeroplane Generations

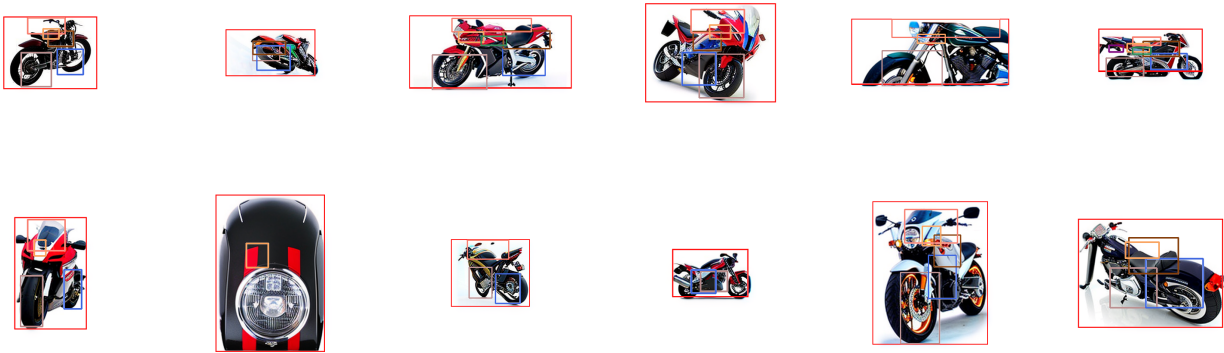


Figure 11. Motorbike Generations



Figure 12. Bicycle Generations

References

115

- [1] Pranav Gupta, Rishubh Singh, Pradeep Shenoy, and Ravikiran Sarvadevabhatla. Olaf: A plug-and-play framework for enhanced multi-object multi-part scene parsing, 2024.

116

117