PARTCONCEPTS: A Unified Mechanism for Fine-Grained Part Localization and Generation

Vaibhav Agrawal[⋆†]
IIIT Hyderabad

Varghese Kuruvilla[⋆]
IIIT Hyderabad

Harsh Rangwani
Adobe Research, Bengaluru

Ravi Kiran Sarvadevabhatla
IIIT Hyderabad

† project lead, ⋆ both authors had equal experimental contribution.

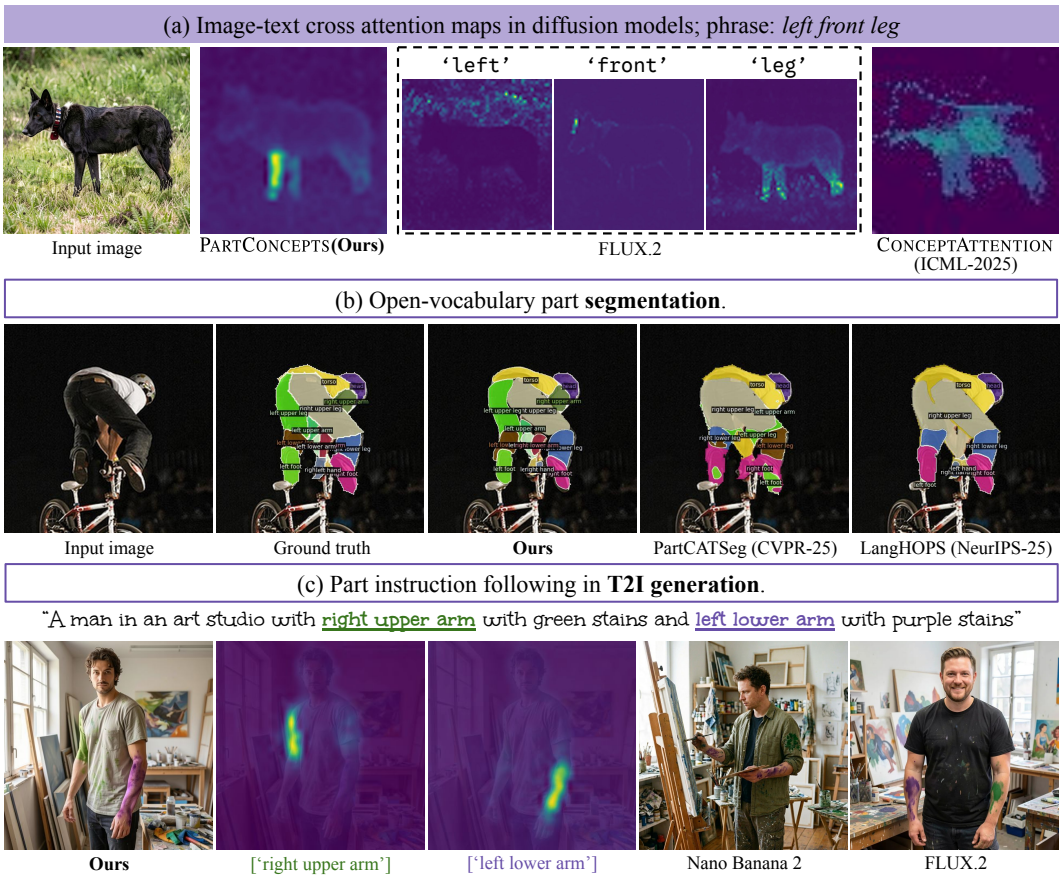

Figure 1: **PARTCONCEPTS.** Cross attention maps of state-of-the-art diffusion models [1] reveal that they lack fine-grained part understanding, shown in (a). To address this, we propose PARTCONCEPTS, a unified mechanism for fine-grained part localization (see b.) and generation (see c.).

Abstract

While text-to-image (T2I) diffusion models exhibit strong semantic disentanglement at the object level, they struggle to localize fine-grained object parts. Although the latent visual features in these models are sufficiently fine-grained, the text–image interaction (cross-attention) fails to effectively exploit this information.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

This results in coarse and ambiguous localization, particularly for spatially distinct components such as 'left' and 'right' limbs. To address this limitation, we introduce **PARTCONCEPTS**, a mechanism that encodes textual part descriptions into learnable PARTCONCEPT tokens. These tokens are trained to selectively attend to their corresponding part regions in the image. To evaluate the part-level understanding of PARTCONCEPT tokens, we first probe its effectiveness for *part segmentation*. We then evaluate whether the improved localisation enables *fine-grained instruction following* in T2I generation. On standard Open-Vocabulary Part Segmentation (OVPS) benchmark, our method outperforms dedicated segmentation baselines, establishing a new state-of-the-art. We further evaluate our method on the more complex Pascal-Part benchmark for *part instance segmentation* which contains fine-grained spatial annotations (e.g. 'left lower arm'), outperforming all baselines significantly (**+15 mIoU**). Finally, our qualitative and quantitative results show that this strong localization of PARTCONCEPT tokens directly enables *fine-grained instruction following* in T2I generation: the textual attributes (e.g., color) *inherently* bind well to corresponding PARTCONCEPT tokens, without explicit attribute control mechanisms. This indicates that PARTCONCEPT tokens compose seamlessly with other text tokens, highlighting their applicability for fine-grained text-based generative control. Code and data are available on our project page.

1 Introduction

Text-to-image (T2I) diffusion models possess strong semantic priors that enable precise generation and editing [2, 3]. Beyond generation, recent work [4] shows that multi-modal Diffusion Transformer (mmDiT) [5, 6] cross-attention maps can serve as competitive object-instance segmentors. These results indicate that the diffusion latent space achieves effective semantic disentanglement at the object level. However, since objects are fundamentally defined by their constituent parts, any system aiming for part-level fine-grained reasoning and precise generative control must rely on semantic part-level representations. This raises a critical question: Does the image-text alignment of diffusion models actually extend to the part-level? Parts are uniquely challenging because quite often a single object contains multiple instances of the same component (e.g., 'left ear' vs. 'right ear'). Distinguishing these requires an understanding of object-centric spatial organization: the relative positioning of parts within the object's own frame of reference [7, 8].

An investigation of the image-text attention maps for state-of-the-art models like FLUX.2 [1] and CONCEPTATTENTION [4] reveals that they are coarse and poorly localized for fine-grained part descriptions (see fig. 1(a)). This poor localization manifests as a failure in downstream instruction following. As shown in fig. 1(c), even state-of-the-art models fail to follow prompts targeting fine-grained parts. **Hence, we conclude that the image-text alignment in diffusion models, while strong at the object level, does not persist at the part level**. To identify the cause of failure, we analyze FLUX.2 and make two key observations:

Observation O1. The latent visual features of diffusion models are semantically very fine-grained, as evidenced by their ability to support highly accurate fine-grained image-to-image pointwise semantic correspondence tasks [9, 10, 7, 11–13], with minimal training supervision.

Observation O2. The text-image interaction (cross-attention) acts as a bottleneck; it fails to leverage the high-resolution latent visual features, resulting in coarse attention maps that cannot distinguish between spatially distinct parts like the 'left' and 'right' limbs, see fig. 1(a).

These observations imply that while the latent features possess sufficient resolution (O1), the text-image interaction acts as a bottleneck (O2). Therefore, our core research question is: **can we improve text aligned part understanding of a T2I model, while preserving its strong priors?** To address this, we introduce PARTCONCEPTS. Given a text prompt, we encode each textual part description (e.g., 'right upper arm') into a single PARTCONCEPT token embedding: this results in a structured token sequence, e.g., 'A photo of an artist with PARTCONCEPT['right upper arm'] ...', which is passed to the mmDiT. During training, this PARTCONCEPT token is optimized to attend to the corresponding part region in the image, providing the fine-grained grounding that base models lack. As we show in our results, our proposed method effectively generalizes to novel part phrases, object classes and prompts beyond what is seen in the training data.

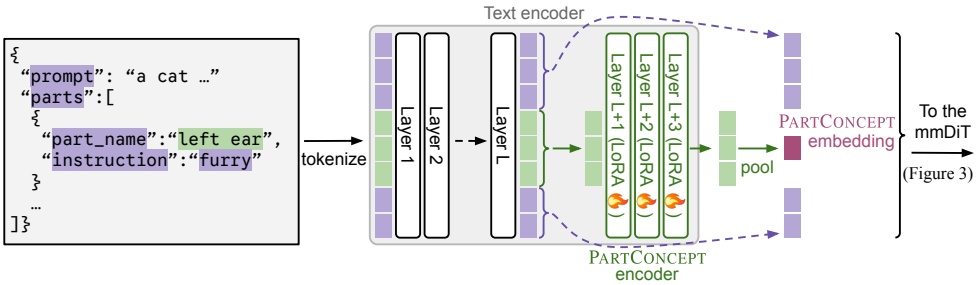

Figure 2: **Encoding textual part descriptions into PartConcepts.** Given an input text prompt (left) with part phrases, our method encodes these textual part phrases (e.g. 'left ear') into PartConcept embeddings, which are trained to attend to corresponding image regions. We extend the text encoding in the base T2I model to encode the PartConcepts: the base T2I model uses intermediate text encoder outputs (at layer L) for conditioning the mmDiT; we use three subsequent text-encoder layers (L+1, L+2, L+3) for forming our PartConcept encoder. We only train a LoRA on these subsequent layers; this design ensures that the output PartConcept does not drift too much from the text embedding space, thus preserving the T2I generative capabilities (see section 4.3).

To evaluate the part-level understanding of PartConcept tokens, we first demonstrate our method's effectiveness for *part segmentation*. We then show that this strong localization directly enables precise part-level attribute binding, thus providing *fine-grained instruction following* in T2I generation. Thus, our PartConcepts provide a unified mechanism for both visual grounding and generative control.

In the setting of *part segmentation*, we extract the part segmentation maps directly from PartConcept attention maps, using only a single forward pass through the mmDiT network; this enables efficient inference without the need for iterative sampling. PartConcepts outperforms dedicated segmentation baselines on the standard Open-Vocabulary Part Segmentation (OVPS) [14] benchmark, establishing state-of-the-art. However, we find that OVPS is too coarse for precise grounding: it typically removes spatial descriptors and collapses distinct instances such as 'left hand' and 'right hand' into a single 'hand' instance. Therefore, we evaluate for *part instance segmentation* using the original Pascal-Part [15] benchmark, which retains these fine-grained part instances and corresponding spatial descriptions. On this more rigorous benchmark, our method significantly outperforms all baselines and establishes a new state-of-the-art.

Finally, we utilize the effective localization of PartConcepts to enable *fine-grained instruction following in text-to-image generation*. We find that our localization-focused training inherently enables the model to bind attributes to specific parts, without a specialized architectural mechanism for attribute control. For example, when provided with a prompt such as 'an artist with PartConcept['right upper arm'] colored green . . . ' (see fig. 1(c)), the model successfully binds the color attribute to the correct part region during the generative process. These results demonstrate that PartConcepts preserve the compositional nature of text space, while maintaining the high-fidelity priors of the base T2I generative model. Our model enables fine-grained instruction following to part descriptions not seen during training, indicating its strong generalization and prior preservation.

2 Related work

Semantic concepts in text-to-image (T2I) diffusion models. Numerous studies have demonstrated that text-to-image (T2I) diffusion models encode strong semantic priors. These priors have also been shown to be effective for downstream tasks such as semantic correspondence [9, 10, 16]. A line of research works introduce learnable prompt tokens, and optimize them to attend to specific visual concepts in the image [17–19, 12]. However, these works require training a separate embedding for every new visual concept, hence not scalable. Further, the learnt prompt tokens are typically not very composable with the text embeddings, hence reducing their applicability for localized attribute binding in T2I generation. A more scalable approach is to use the open vocabulary priors of the T2I models for learning visual concepts. Recent work, ConceptAttention [4] proposes a training free framework for interpreting text-to-image multi-modal diffusion transformers (mmDiT), by extracting similarity maps of diffusion features with *textual concept embeddings*. While these maps demonstrate strong object-level semantics, they lack fine-grained part-level understanding

(see fig. 1(a)), consistent with our observations. We attribute this limitation primarily to the weak alignment between textual concepts and visual features in the text-image interface of these models, and propose an attention-based supervision strategy to mitigate this.

Part instance segmentation in images is a long standing problem in computer vision [20–26], with various datasets proposed for this task [15, 27–29]. Among these, Pascal-Part [15] stands out for its fine-grained part descriptions (e.g. 'left ear', 'right front upper leg'). Open-Vocabulary Part Segmentation (OVPS) [14] is the standard benchmark used by recent methods [30–35]; however, OVPS is too coarse, as it removes spatial descriptors and collapses distinct instances such as 'left hand' and 'right hand' into a single 'hand' instance (see Appendix H). Among recent works, LangHOPS [35] parses object hierarchies using an MLLM to generate queries which are decoded into part-level segmentation masks. PartCATSeg [34] introduces a cost aggregation framework with a compositional loss and structural guidance using DINO [36]. In contrast, our work leverages strong priors of generative models using a simple architecture to achieve fine-grained open-set part segmentation.

Part level instruction following in T2I generation. Part level instruction following has been explored in prior work [37], which uses self attention clustering followed by attribute injection through cross attention. However, it assumes image-text cross attention to be highly localized, which is violated in many cases, especially for fine-grained part descriptions. While there are works which focus on learning part-level concepts from images, and compose them for creative generations [18, 38], they do not support text-based instruction following, hence not directly relevant. Recent work [39] uses learnable part concepts for localized image editing. However, it requires an embedding optimization procedure for every new part concept, hence not scalable. Further, it can only enable editing in a provided reference image. In contrast, our method directly enables instruction following in text-to-image generation, which is challenging due to absence of a reference structure.

3 Method

Given an input text prompt containing part phrases (e.g., 'left ear', fig. 2), our goal is to ensure that these phrases attend strongly to their corresponding image regions, thereby enabling both part localization and fine-grained instruction following in T2I generation.

In standard FLUX.2 cross-attention maps (fig. 1a), multi-word part phrases are split into discrete tokens that attend diffusely across disparate regions. Spatial descriptors (e.g., 'left', 'right', etc.) are especially dispersed, preventing the overall part description from localizing to a coherent visual region.

To address this, we encode the text embeddings of each part phrase into a single PARTCONCEPT embedding of matching dimensionality (fig. 2). Replacing discrete part-phrase tokens with their corresponding PARTCONCEPT token yields a structured sequence, which is fed into the multi-modal diffusion transformer (mmDiT) alongside noisy visual tokens (fig. 3). To prevent PARTCONCEPT embeddings from drifting away from the text manifold, we initialize the PARTCONCEPT encoder using successive layers of the base text encoder and optimize only a LoRA [40] adapter. As shown in section 4.4, this parameter-efficient design preserves the base T2I model's generative prior.

The localization objective. To localize PARTCONCEPT tokens to correct image regions, we optimize their cross-attention maps using a Dice segmenta-

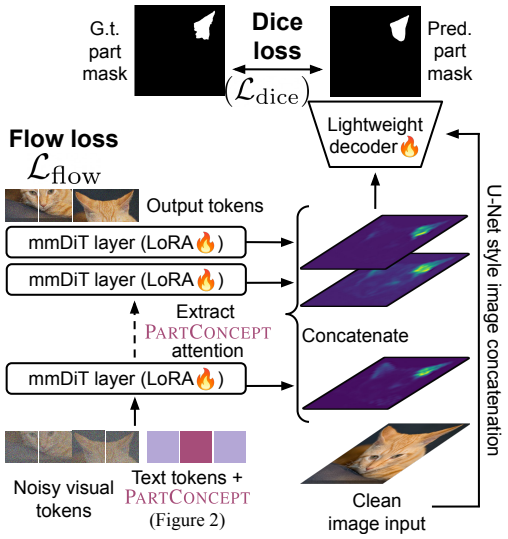

Figure 3: **The localization objective.** We pass the interleaved text and PARTCONCEPT tokens see fig. 2) through the text branch of the mmDiT model. During forward pass through the mmDiT, we extract layer-wise cross attention maps of the PARTCONCEPT tokens. These attention maps, along with clean input image are decoded into precise part segmentation maps using a multi-scale CNN decoder, trained using dice loss. **The generative objective.** To preserve the generative capability of the T2I model, we also include the flow loss (denoising MSE), enabling fine-grained *part-level instruction following* in T2I generation.

tion loss [41]. While PARTCONCEPT cross attention maps provide strong semantic localization, directly converting them into dense segmentation masks faces fundamental limitations: First, diffusion models operate over noisy latents, meaning the resulting cross attention maps inherently lack access to uncorrupted, fine-grained image evidence. Second, VAE latent compression restricts attention maps to a coarse spatial grid, precluding pixel-accurate contour prediction. Finally, relying on raw attention maps necessitates brittle heuristics, including layer selection and manual thresholding. We provide a detailed ablation in Appendix A.1.

To resolve these bottlenecks, we employ a lightweight segmentation decoder (fig. 3, top-right). The decoder learns to fuse concatenated multi-layer PARTCONCEPT attention maps while being explicitly conditioned on the clean RGB image across multiple spatial scales. This architecture couples the coarse semantic localization encoded in the diffusion attention maps with the sharp, uncorrupted edge cues of the clean image, yielding precise, end-to-end part segmentations.

During part-segmentation inference, given an input image and query part phrases, we first map the phrases into PARTCONCEPT tokens (fig. 2) and encode the image into diffusion latents with a fixed noise timestep (details in Appendix A.2). A single mmDiT forward pass then extracts the PARTCONCEPT cross-attention maps, which the decoder processes into final part masks (fig. 3).

Preserving the generative capability. While the localization objective grounds PARTCONCEPT tokens to their respective visual regions, directly fine-tuning the model on segmentation alone risks degrading its core synthesis capabilities. To retain the base model's generative prior, we include the flow matching objective (velocity prediction MSE) alongside the localization loss. The joint training objective is formulated as: $\mathcal{L} = \mathcal{L}_{\text{dice}} + \lambda \cdot \mathcal{L}_{\text{flow}}$, where λ governs the trade-off between localization precision and visual fidelity of generation outputs, as analyzed in section 4.4.

Training dataset. To train our model, we leverage the Pascal-Part [15] dataset, chosen for its fine-grained, instance-level spatial annotations of object parts. However, naively training modern diffusion models on Pascal-Part poses a domain alignment challenge: its legacy images feature low resolutions and skewed aspect ratios that can degrade the high-fidelity visual priors of the base T2I model. To overcome this, we design an automated restoration pipeline using FLUX.2 that upsamples and outpaints the images while strictly preserving the underlying part annotations. Training on this enhanced benchmark, termed "Pascal-Part SR" (Super-Resolution), ensures compatibility with the visual quality of modern generative models [1] (see Appendix B for details).

Attribute binding. Beyond strong attention localization, accurate *part-level instruction following* requires that the PARTCONCEPT tokens semantically compose well with surrounding textual attributes. For instance, in a prompt specifying distinct textures or colors for the 'right upper arm' versus 'left lower arm' (Fig. 1c), the model must correctly bind each attribute with its corresponding PARTCONCEPT token. Rather than introducing explicit binding mechanisms, we foster compositionality at the data level: we augment training prompts with localized part captions extracted by querying a VLM [42] on cropped part regions (see Appendix C for details). Remarkably, this prompt augmentation implicitly enables robust attribute-part binding during generation, without any explicit architectural mechanisms.

4 Experiments

We first evaluate our method's effectiveness on part-level localization using the standard *part segmentation* [14] (section 4.1) and *part instance segmentation* [15] benchmarks (section 4.2). We then show that the proposed PARTCONCEPTs' strong localization directly enables fine-grained *part-level instruction following* in T2I generation (section 4.3).

4.1 Open vocabulary part segmentation (OVPS) [14]

Implementation and training details. We use FLUX.2 Klein Base 9B [1] as the base model for all experiments in this paper. For the PARTCONCEPT encoder, we train a LoRA (rank 128) over the text-encoder layers (see fig. 2). Additionally, we train a smaller LoRA (rank 4) for the mmDiT, following standard practice. We train the models until convergence, at batch size 16, using default optimizer settings from HuggingFace Diffusers [43]. We train our models on the respective train splits from the OVPS benchmark, following standard evaluation protocols.

Table 1: **Zero-shot evaluation results.** Our method outperforms dedicated segmentation baselines on Open-Vocabulary Part Segmentation (OVPS) [14] benchmark, establishing state-of-the-art.

Method	ADE20K [14]			PPS-116 [14]			PartImageNet [27]		
	seen	unseen	h-IoU	seen	unseen	h-IoU	seen	unseen	h-IoU
VLPart [44]	*	*	*	42.6	18.7	26.0	*	*	*
ZSSeg+ [45]	43.2	27.8	33.9	54.4	19.0	28.2	*	*	*
CLIPSeg [46]	38.2	30.9	34.2	48.9	27.5	35.2	53.9	37.2	44.0
CAT-Seg [47]	33.8	25.9	29.3	43.8	27.7	33.9	47.3	35.1	40.3
PartCLIPSeg [48]	38.4	38.8	38.6	50.0	31.7	38.8	56.3	51.7	53.9
PartGLEE [31]	51.3	35.3	41.8	57.4	27.4	37.1	*	*	*
PartCATSeg [49]	53.1	47.2	50.0	57.5	44.9	50.4	73.8	71.5	72.7
LangHOPS [35]	49.3	49.7	49.5	59.2	46.5	52.1	71.9	**73.7**	72.8
Ours	**53.4**	**61.9**	**57.3**	**65.0**	**49.7**	**56.3**	**75.4**	70.6	**72.9**

Datasets and evaluation. Following OVPS [14], we evaluate two setups: *zero-shot*, where train and test categories do not overlap, and *cross-dataset*, where evaluation is performed on a dataset different than the one used for training, hence testing OOD generalization. In the context of *zero-shot* part segmentation, we use Pascal-Part-116 (PPS-116) [14], ADE20K-Part-234 [14] and PartImageNet [27] datasets. Following [34], we evaluate *cross-dataset* transfer for 1) PartImageNet (training) to PPS-116 (evaluation), and 2) PartImageNet (OOD), a split of the PartImageNet dataset which was proposed originally for few-shot part segmentation [27]. Similar to [35], we use the oracle-object setting [14], where ground truth object segmentation and class are provided. Following prior work [14, 33–35], we use the IoU metric, reported for both seen and unseen categories separately, as well as their harmonic mean (h-IoU). Further details are presented in Appendix G.

Table 2: **Cross-dataset evaluation results.** We consider two settings: PartImageNet (OOD) [27] and PartImageNet (train) → PPS-116 [14] (eval).

Method	PartImageNet (OOD) [27]	PartImageNet → PPS-116 [14]
	Unseen	Unseen
CLIPSeg [46]	55.86	14.87
CAT-Seg [47]	39.12	12.50
PartCLIPSeg [48]	59.16	19.86
PartCATSeg [49]	66.15	22.88
Ours	**67.30**	**30.00**

Baseline comparison. We compare against standard part segmentation baselines; recent works include PartCATSeg [34] and LangHOPS [35]. Evaluation results are presented in table 1 (*zero-shot*) and table 2 (*cross-dataset*). Our method outperforms dedicated segmentation baselines in almost all categories, establishing state-of-the-art. We visualize segmentation results in fig. 7.

4.2 Fine-grained part *instance* segmentation [15]

While our method is competitive on (OVPS) [14] benchmark, we find that the OVPS suite is too coarse for precise grounding: it typically removes spatial descriptors and collapses distinct instances such as 'left hand' and 'right hand' into a single 'hand' instance (see fig. 20 and Appendix H). Therefore, we evaluate for part *instance* segmentation by training our method (and baselines) on the original Pascal-Part [15] benchmark, which retains these fine-grained part instances and corresponding spatial descriptions. On this more rigorous benchmark, our method significantly outperforms dedicated perception baselines and establishes a new state-of-the-art. We visualize segmentation results in fig. 7.

Table 3: **Fine-grained part *instance* segmentation**, with spatial identification of part instances (e.g. 'left lower arm', 'right upper arm').

Method	Pascal-Part (org.) [15]		
	seen	unseen	h-IoU
PartCATSeg [49]	33.2	35.5	34.3
LangHOPS [35]	38.4	38.7	38.5
Ours	**55.2**	**54.2**	**54.7**

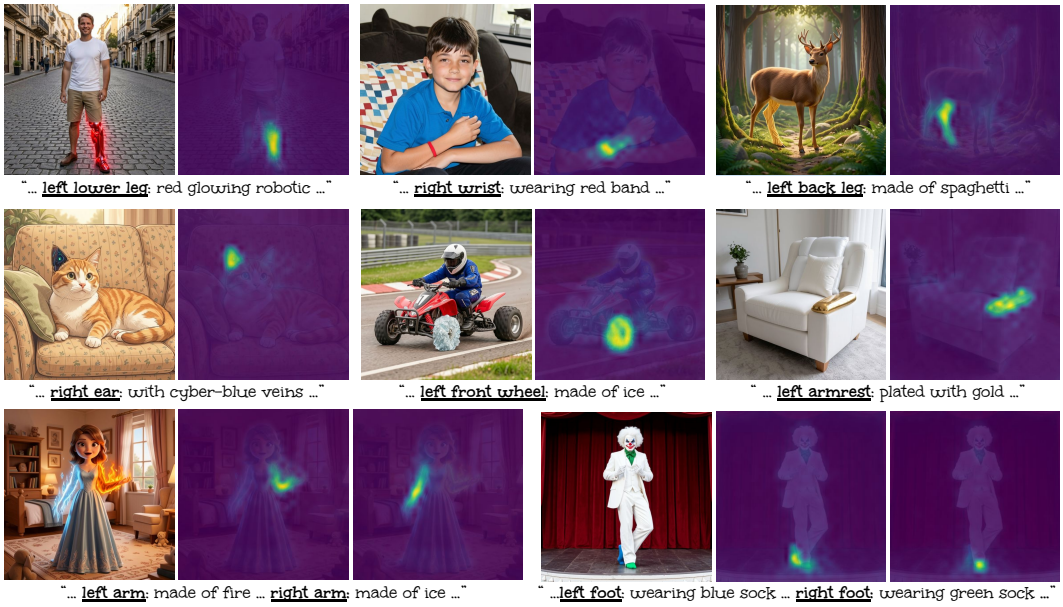

Figure 4: **Qualitative results for *part-level instruction following*, with PARTCONCEPT attention maps.** Our method is able to disambiguate spatial identifiers in part descriptions, (e.g., 'left front wheel'). Notably, PARTCONCEPTs generalize to novel part phrases; specifically, ***armrest, left front wheel* and *wrist* are not seen in our training data**, indicating strong generalization. Even in case of multiple parts, the attributes are correctly bound to respective PARTCONCEPTs; this demonstrates that PARTCONCEPT embeddings preserve the compositional nature of text embeddings.

4.3 Fine-grained instruction following

Implementation and training details. We use the same settings as described in section 4.1. Additionally, to preserve the generative capability of the base T2I model, we set the loss balancing coefficient $\lambda = 0.25$ (section 3). We train this model on the Pascal-Part SR dataset, thus preserving the photorealistic prior of the generative model.

Baselines. We compare our method against our base T2I model FLUX.2 Klein Base 9B [1], Nano Banana 2 [50, 51], and FLUX.1 [dev] [52].

Quantitative evaluation. We primarily evaluate our method (and baselines) for **attribute localization**, i.e., *how strongly are the attributes bound to the intended parts*? We construct an evaluation dataset of part-level instructions: given a datapoint from the Pascal-Part dataset, we sample a part label and couple it with a randomized instruction, such as 'stain with purple color'. The resulting text prompts follow the template 'a photo of `{class_name}`, with `{instruction}`'. We construct evaluation examples with one and two part-level instructions per prompt. Part-masks for generated images are obtained using our PARTCONCEPTS segmentation model (section 4.1), followed by manual

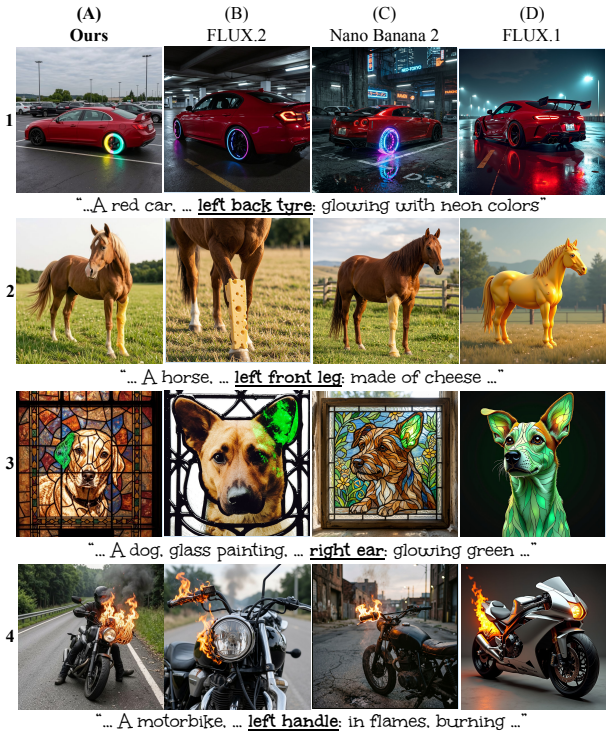

Figure 5: **Qualitative comparisons.** Our method exhibits strong spatial localization, while baselines frequently struggle, failing to disambiguate spatial identifiers in part descriptions (e.g., 'left back wheel', 'right ear', etc.).

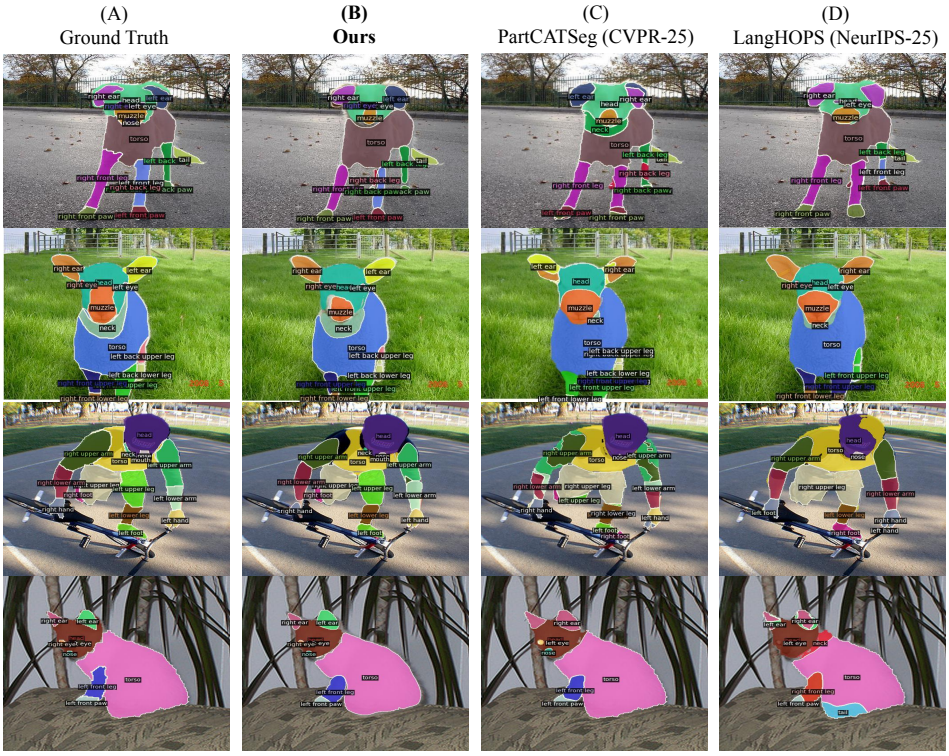

Figure 7: **Qualitative comparison.** Our method outperforms all baselines on various benchmarks, establishing state-of-the-art. Notably, we are significantly better for *part instance segmentation* [15], effectively disambiguating spatially distinct part instances like 'left lower arm' and 'right upper arm'.

refinement. We use these part-masks to evaluate localized attribute correctness using a VLM-as-a-judge framework, which has been used for evaluating compositional T2I generation in prior work [53, 54]. Specifically, the VLM is provided with an annotated image with the target part boundary highlighted, and asked to provide a score according to the following rubric:

- 1.0 if the attribute is not present in indicated region.
- 2.0 to 4.0 if the attribute is present in the indicated region, but also leaks to other parts/regions outside the part boundary. A score of 2.0 indicates severe leakage onto other parts, while 4.0 indicates minor/slight leakage.
- 5.0 if the attribute is present in the indicated part AND is localized well.
- A score of 0.0 is automatically assigned if the part is not visible in the generated image (empty-mask).

We perform this evaluation using Qwen3-VL-8B-Instruct [42] and Gemini 3.5 Flash [51], with results reported in table 4. FLUX.1 [dev] applied the requested attribute to the entire object for most cases (see fig. 5 D2-3), and hence we removed it from this evaluation. We also performed an A/B user study (see fig. 6), where users were asked to compare between our method's output and a randomly chosen baseline using similar instructions as the above rubric. More details about the user study are provided in Appendix J. The evaluations consistently highlight that our method enables more accurate part-level instruction following compared to baselines.

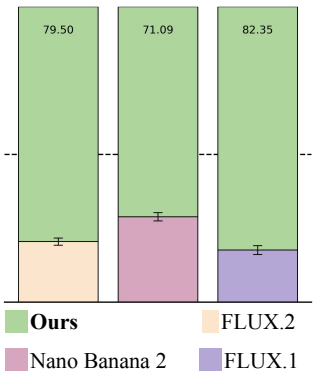

Figure 6: **User study** evaluating localization of attributes to respective parts.

Qualitative evaluation. As shown in fig. 4, the PARTCONCEPT strongly localizes to correct part regions, shown by the sharp attention maps, while preserving T2I generation fidelity. Our PARTCONCEPT encoder exhibits strong generalization to novel part phrases (e.g., ***armrest***, ***left front wheel***, ***wrist*** **(fig. 4) and** ***handle***, ***tyre*** **(fig. 5) are not present in our training data**).

Table 4: **VLM-as-judge** evaluation of attribute localization.

Model	Qwen [42]	Gemini [51]
Nano Banana 2 [50, 51]	2.49	1.73
FLUX.2 Klein [1]	2.34	1.60
Ours	**3.52**	**2.45**

Further, our method effectively leverages the T2I generative prior to generalize to arbitrary attributes: none of the shown part attributes in figs. 4 and 5 were seen in our training data. Notably, our method performs correct attribute binding even in multi-part scenarios (third row, fig. 4), without any explicit mechanisms for attribute binding. This demonstrates that the PARTCONCEPT token preserves compositional nature of the text space, and thus behaves *similarly* to other text tokens. Fig. 5 (B,C) shows that FLUX.2 and Nano Banana 2 often fail to generate the intended part in some cases, or fail to correctly bind the attribute. In FLUX.1 generations the attribute is highly diffused, and not restricted to small part regions (fig. 5 D).

4.4 Ablations

We validate our approach through several ablative experiments. First, the flow loss weight (λ) balances localization and denoising objectives during training (see fig. 8). We find that λ dictates a tradeoff between generation fidelity and attribute binding (see fig. 8); low values degrade the generative prior, causing visual artifacts, while high values reduce localization pressure, leading to attribute leakage.
Second, LoRA finetuning for PARTCONCEPT encoder (see fig. 2) is critical; full fine-tuning causes significant visual artifacts (highlighted with red boxes in fig. 9(B)), as PARTCONCEPT embedding drifts too much from the text space, disrupting the T2I prior. Third, our VLM-based prompt augmentation is essential for attribute binding; without it, the training lacks incentive to associate specific attributes with PARTCONCEPTs, as shown in fig. 9(C).
Finally, we find batch size significantly impacts localization performance. Unlike standard FLUX.2 [1] fine-tuning scripts [43] that use small batches, our localization loss trajectories significantly improve with larger batch sizes (fig. 10, we also expect improvements with scaling). We further ablate upon additional components of the trained PARTCONCEPTS model in Appendix A.



Figure 8: **Effect of increasing flow loss weight** (λ). Low values lead to visual artifacts (left), while higher values cause attribute leakage.

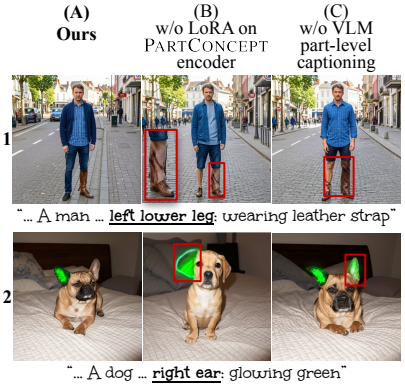

Figure 9: **Ablations. B.** LoRA on the PARTCONCEPT is crucial for preserving generation fidelity. **C.** Training prompt augmentation using VLM captioning improves attribute binding.

5 Limitations

Despite its effectiveness, PARTCONCEPTS faces two primary limitations. First, scaling beyond four part-attribute pairs occasionally causes attribute leakage (see Appendix D), even when spatial localization is accurate. This likely stems from a distributional shift, as prompts with more than four attributes are out-of-distribution (OOD) relative to our training data. Second, while we provide a unified mechanism for segmentation and T2I part-level control, the best-performing loss weights (λ) differ for each task. This reflects a slight trade-off between discriminative grounding and generative prior preservation. Future research should try to mitigate this gap by designing the model in a way that it leads to optimal discriminative localization capabilities without having any tradeoff on generative prior preservation.

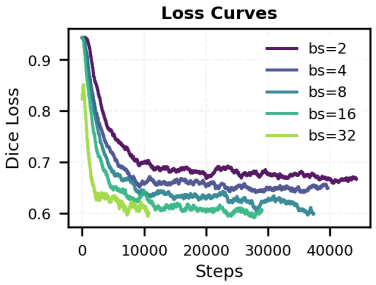

Figure 10: **Scaling batch size.** The localization loss trajectories respond positively to increasing batch sizes.

6 Conclusion

This work stems from observing that while T2I cross attention maps are strongly disentangled for objects, this semantic clarity collapses at the part-level, particularly for spatially distinct part instances (e.g., 'left front leg' vs. 'right front leg'). We identify the bottleneck as text-image interaction (cross attention) rather than the visual feature resolution. Hence, we propose a lightweight mechanism to correct this: encoding part phrases (e.g. 'left front leg') into a compact representation: the PARTCONCEPT tokens, trained for spatial grounding. Our approach effectively leverages the base T2I's generative priors to achieve state-of-the-art performance on various *part (instance) segmentation* benchmarks, outperforming dedicated perception baselines. We further show that this effective localization directly enables robust attribute binding and *fine-grained instruction following* in T2I generation. These results demonstrate the effectiveness of the proposed approach for improving part-level understanding of generative models. We believe that the proposed attention based supervision can be extended as a general post-training mechanism for T2I diffusion models to strengthen specific textual concepts underrepresented during pretraining. We leave this exploration for future work.

7 Acknowledgements

We thank Medha Sen, Amal Joseph and other members of CVIT, IIIT Hyderabad for generously prioritizing compute access for this project, especially during the last days before deadline. We are grateful to Aishwarya Agarwal, Ayan Kashyap, and Rishubh Parihar for helpful discussions, and to Darshana and Medha for help in reviewing the manuscript. This project was supported by a grant from Adobe Research.

References

[1] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

[2] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[3] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7877–7888, 2025.

[4] Alec Helbling, Tuna Han Salih Meral, Ben Hoover, Pinar Yanardag, and Duen Horng Chau. Conceptattention: Diffusion transformers learn highly interpretable features. *arXiv preprint arXiv:2502.04320*, 2025.

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[6] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[7] Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. Telling left from right: Identifying geometry-aware semantic correspondence. 2023.

[8] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-guided spherical maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19521–19530, 2024.

[9] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36: 45533–45547, 2023.

[10] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ypOiXjdfnU.

[11] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023.

[12] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *Advances in Neural Information Processing Systems*, 36:8266–8279, 2023.

[13] Chaofan Gan, Yuanpeng Tu, Xi Chen, Tieyuan Chen, Yuxi Li, Mehrtash Harandi, and Weiyao Lin. Unleashing diffusion transformers for visual correspondence by modulating massive activations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=s3MwCBuqav.

[14] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[15] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[16] Olaf Dünkel, Thomas Wimmer, Christian Theobalt, Christian Rupprecht, and Adam Kortylewski. Do it yourself: Learning semantic correspondence from pseudo-labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5834–5844, 2025.

[17] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023.

[18] Junyu Liu, R Kenny Jones, and Daniel Ritchie. Partcomposer: Learning and composing part-level concepts from single-image examples. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025.

[19] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Xingzhe He, Hossam Isack, Abhishek Kar, Helge Rhodin, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised keypoints from pretrained diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22820–22830, June 2024.

[20] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC'04)*, pages 779–788. The British Machine Vision Association (BMVA), 2004.

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[22] Andrea Vedaldi, Siddharth Mahendran, Stavros Tsogkas, Subhransu Maji, Ross Girshick, Juho Kannala, Esa Rahtu, Iasonas Kokkinos, Matthew B Blaschko, David Weiss, et al. Understanding objects in detail with fine-grained attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3622–3629, 2014.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[24] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5485–5494, June 2021.

[25] Xiangtai Li, Shilin Xu, Yibo Yang, Haobo Yuan, Guangliang Cheng, Yunhai Tong, Zhouchen Lin, Ming-Hsuan Yang, and Dacheng Tao. Panoptic-partformer++: A unified and decoupled view for panoptic part segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):11087–11103, 2024.

[26] Rishubh Singh, Pranav Gupta, Pradeep Shenoy, and Ravikiran Sarvadevabhatla. Float: Factorized learning of object attributes for improved multi-object multi-part scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1445–1455, 2022.

[27] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022.

[28] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[29] Xiao Li, Yining Liu, Na Dong, Sitian Qin, and Xiaolin Hu. Partimagenet++ dataset: Scaling up part-based models for robust recognition. In *European Conference on Computer Vision*, pages 396–414. Springer, 2024.

[30] Tai-Yu Pan, Qing Liu, Wei-Lun Chao, and Brian Price. Towards open-world segmentation of parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15392–15401, 2023.

[31] Junyi Li, Junfeng Wu, Weizhi Zhao, Song Bai, and Xiang Bai. Partglee: A foundation model for recognizing and parsing any objects. In *European Conference on Computer Vision*, pages 475–494. Springer, 2024.

[32] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Advances in Neural Information Processing Systems*, 36:21429–21453, 2023.

[33] Jiho Choi, Seonho Lee, Seungho Lee, Minhyun Lee, and Hyunjung Shim. Understanding multi-granularity for open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 37:137161–137189, 2024.

[34] Jiho Choi, Seonho Lee, Minhyun Lee, Seungho Lee, and Hyunjung Shim. Fine-grained image-text correspondence with cost aggregation for open-vocabulary part segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9782–9793, 2025.

[35] Yang Miao, Jan-Nico Zaech, Xi Wang, Fabien Despinoy, Danda Pani Paudel, and Luc Van Gool. Langhops: Language grounded hierarchical open-vocabulary part segmentation. *arXiv preprint arXiv:2510.25263*, 2025.

[36] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021.

[37] Harsh Rangwani, Aishwarya Agarwal, Kuldeep Kulkarni, R Venkatesh Babu, and Srikrishna Karanam. Composing parts for expressive object generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13209–13219, 2025.

[38] Kam Woh Ng, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Partcraft: Crafting creative objects by parts. In *European Conference on Computer Vision*, pages 420–437. Springer, 2024.

[39] Aleksandar Cvejic, Abdelrahman Eldesokey, and Peter Wonka. Partedit: fine-grained image editing using pre-trained diffusion models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[41] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[42] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[43] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[44] Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Going denser with open-vocabulary part segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15453–15465, 2023.

[45] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open vocabulary semantic segmentation with pre-trained vision-language model. *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2022.

[46] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, June 2022.

[47] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4123, 2024.

[48] Jiho Choi, Seonho Lee, Seungho Lee, Minhyun Lee, and Hyunjung Shim. Understanding multi-granularity for open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 37:137161–137189, 2024.

[49] Jiho Choi, Seonho Lee, Minhyun Lee, Seungho Lee, and Hyunjung Shim. Fine-grained image-text correspondence with cost aggregation for open-vocabulary part segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9782–9793, 2025.

[50] Google. Nano banana 2: Combining pro capabilities with lightning-fast speed. https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/, February 26 2026. Accessed: April 29, 2026.

[51] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[52] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[53] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, et al. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024.

[54] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

[55] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11: 285–296, 1975.

[56] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[57] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

A Analyzing the trained PARTCONCEPTS model

In this section we analyze our trained PARTCONCEPTS model, studying the impact of each trainable component (the segmentation decoder, PARTCONCEPT encoder and LoRA [40] on the mmDiT), choice of inference-time noise levels for localization inference, and conclude with a discussion about how our approach significantly enhances the capabilities of the base T2I model.

A.1 Role of the segmentation decoder

Directly using PARTCONCEPT attention maps for *part instance segmentation* is challenging for two reasons. First, the attention maps in mmDiT are computed using the compressed latents, hence they lack the resolution required for dense part mask prediction at the original image resolution. Second, directly using attention maps for downstream segmentation is quite brittle, as it necessitates extra hyperparameters for attention map thresholding and layer selection.

To demonstrate this, we set up a simple baseline, where we average the layer-wise attention maps obtained after forward pass, average them and apply bilinear upsampling to get a single map, of the same dimensions as the input image. Finally, we apply the adaptive OTSU thresholding [55] to get the part segmentation masks. As illustrated in fig. 11, while this baseline is able coarsely localize the part regions, it fails to match the performance of the full model with the segmentation decoder. This is because, while the PARTCONCEPT attention maps are coarsely localized to the correct region, they lack the fine-grained resolution necessary for dense segmentation prediction; the segmentation decoder mitigates this issue by combining the coarse localization of PARTCONCEPT attention maps with the fine details from input image to output precise part segmentation maps.

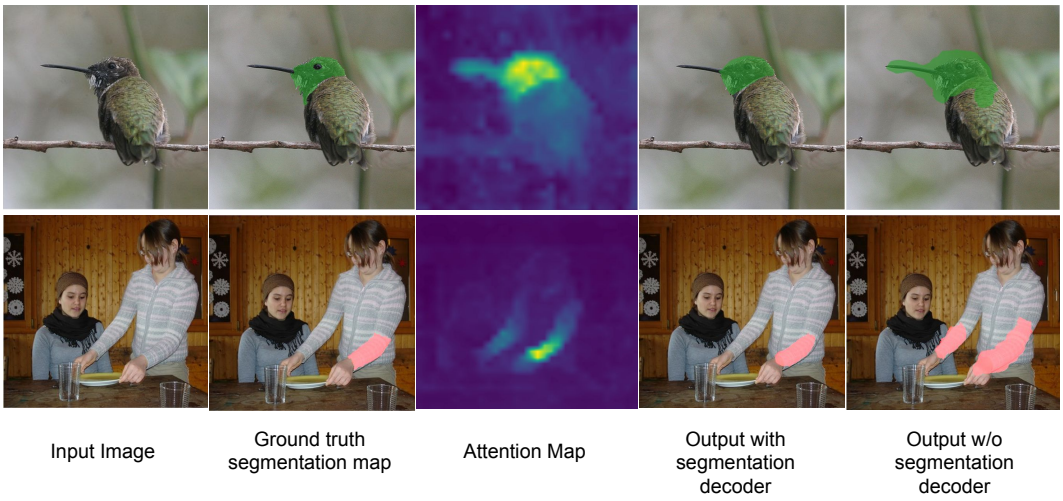

Figure 11: **Impact of segmentation decoder on *part instance segmentation*.** While PARTCONCEPT attention maps are able to coarsely localize the part regions, using them directly for segmentation fails to match the performance of the full model with segmentation decoder. This is because while the PARTCONCEPT attention maps are coarsely localized to the correct region, they lack the fine-grained resolution necessary for dense segmentation prediction; segmentation decoder mitigates this issue by combining the coarse localization of PARTCONCEPT attention maps with fine details from the input image to output precise part segmentation maps.

A.2 Impact of inference timestep

For inferring part segmentation maps, we encode the input image into diffusion latent space and add a fixed amount of noise to it. We investigate the sensitivity of our framework to the noise timestep t used during inference ($t = 0$ is zero noise (clean image), $t = 1000$ is maximum noise). We perform this analysis on the OVPS Pascal-Part split [14]. As shown in table 5, the model exhibits a stable performance across the mid-range of the noise schedule ($t = 200$ to $t = 600$), with best performance at $t = 400$.



Figure 12: **Comparison with base T2I model, and impact of each trainable component on the generative task.** We present A. our method's output, B. base T2I model (FLUX.2) outputs, C. output without LoRA on mmDiT (section A.4), and D. output without having the PARTCONCEPT encoder (section A.3). We find that the base T2I model (FLUX.2 Klein 9B [1]) demonstrates lack of spatial part instance understanding, demonstrated by inability to correctly follow part-level instructions (B); our approach significantly enhances this understanding, thus enables finer text-based generative control (section A.5). We find that while removing the LoRA on mmDiT does not have a significant impact (C), PARTCONCEPT encoder is crucial for binding the attributes to correct part regions (D). ***All comparisons presented in this figure use the same starting latent.***

A.3 Impact of PARTCONCEPT encoder

As shown in fig. 12(D), the PARTCONCEPT encoder is essential for *part-level instruction following* in T2I generation; without it, the model fails to bind attributes to the correct part region. This demonstrates that PARTCONCEPT encoder is the major bridge that *aligns* the textual part-level phrases to the visual features of the diffusion model.

A.4 Role of the LoRA on mmDiT

Following standard finetuning scripts provided in the HuggingFace Diffusers [43] framework, we use a small LoRA [40] of rank 4 on the mmDiT. However, as illustrated in fig. 13, this trainable component seems to have minimal impact on the final part segmentation maps. Similarly, for the

Table 5: **Impact of the inference timestep t on segmentation performance.** The model exhibits a stable performance across the mid-range of the noise schedule ($t = 200$ to $t = 600$), with best performance at $t = 400$.

Timestep (t)	mIoU Base	mIoU Novel	hIoU
200	63.90	49.19	55.50
400	**65.04**	**49.70**	**56.30**
600	64.87	49.47	56.10
800	61.88	45.42	52.30

generative task, the model binds attributes correctly to the respective parts, even in absence of LoRA on the mmDiT fig. 12(C).



Figure 13: **Impact of trainable mmDiT LoRA parameters on *part instance segmentation*.** We find that the mmDiT LoRA parameters have minimal impact on the final part segmentation accuracy.

A.5 Conclusion: enhancing the base T2I model capabilities.

To conclude this analysis, we present a comparison with the base T2I model used for PARTCONCEPTS, FLUX.2 Klein 9B [1], see fig. 12(B). Clearly, the base T2I model fails to bind attributes to the correct parts: when prompted to apply gemstone texture (fig. 12 1-2B) separately to the 'left front leg' and 'right front leg' of the horse, it applies the texture to the right front leg in both cases. Similar case with 'left hand' and 'right hand' in fig. 12 3-4B, where the model fails to associate the bandage attribute to the correct hand. These results indicate an absence of spatial part instance understanding in the base T2I model. Our approach significantly enhances this understanding, hence enables finer text-based generative control.

B Pascal-Part SR: Enhancing Pascal-Part [15]

Pascal-Part [15] is a state-of-the-art dataset for part instance segmentation. It is especially useful because of its fine-grained labels, with spatial identification of part instances (e.g., 'left front upper leg', 'right lower arm', etc.). Due to its granularity, we utilize it in our model training for *fine-grained instruction following* (see section 4.3). However, images of this dataset are low-resolution and have irregular aspect ratios; training on such low quality images affects the high-fidelity generation capabilities of the base T2I model.

To mitigate these issues while maintaining the integrity of the original dense labels, we propose an automated image-enhancement pipeline leveraging the restoration capabilities of FLUX.2 [1]. As a pre-processing step, we first isolate individual object instances by cropping them from the original multi-object Pascal-Part images, applying a marginal spatial padding to preserve local context. As illustrated in fig. 16(a), the subsequent pipeline consists of two stages: (1) Super-Resolution (SR) to increase the pixel density of the raw training data, followed by (2) symmetric outpainting to rectify skewed aspect ratios.

A critical design requirement of this pipeline is the preservation of dense, part-level annotations across both transformations. During the SR stage, which maintains the original aspect ratio, we apply an identical scaling factor to the segmentation masks to ensure pixel-perfect alignment with the upscaled features. In the subsequent outpainting stage, we employ a symmetric padding strategy, extending the canvas boundaries equally relative to the image center. This approach ensures that the original image content—and its corresponding dense annotations—remains undisturbed, allowing for a seamless mapping to the new coordinate space.

While the proposed pipeline successfully enhances the majority of Pascal-Part [15] images, certain edge cases, specifically images with extreme low resolution or severe aspect ratio skew present challenges. **Structural detail degradation in SR (fig. 16(b)):** For images below a certain critical resolution, the super-resolution fails to recover fine-grained structural details, leading to hallucinations. To maintain the high fidelity required for model training, we implement a minimum resolution threshold, filtering out samples that lack sufficient initial information for high-quality restoration. **Annotation discontinuity in outpainting (fig. 16(c)):** Because the outpainting process generates new visual content beyond the original image boundaries, these newly synthesized regions lack corresponding ground-truth labels. This results in "semantic cuts" or abrupt boundaries where the original dense part annotations terminate. We accept this trade-off to achieve the uniform aspect ratios necessary for preserving generative capabilities of the base T2I.

The resulting Pascal-Part SR dataset consists of 15493 enhanced object-instance images, out of 20173 total valid instances in the original Pascal-Part dataset. Training on this enhanced dataset ensures compatibility with the high visual quality of state-of-the-art generative models [1].

C VLM-based part-level captioning

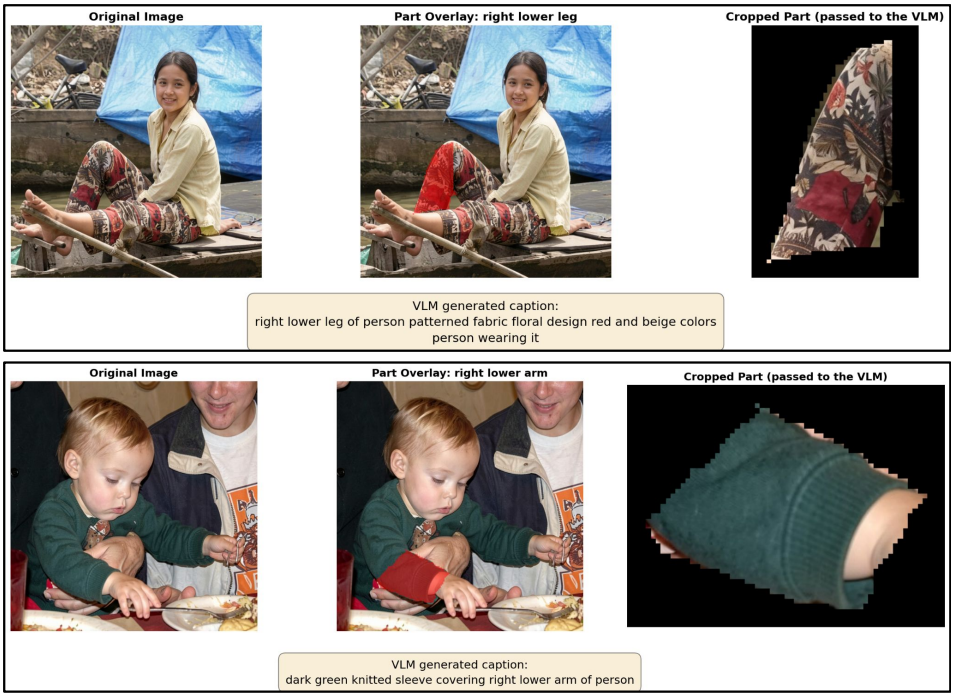

Figure 14: **VLM-based part captioning.** To ensure that the PARTCONCEPT tokens compose well with other text tokens, we augment the training prompts with part-level captions: we pass cropped out part regions from the training images through a VLM [56], which generates suitable captions describing the part regions.

To ensure that the PARTCONCEPT tokens compose well with other text tokens, we augment the training prompts with part-level captions: we pass cropped out part regions from the training images through a VLM [56], which generates suitable captions describing those parts (see fig. 14).

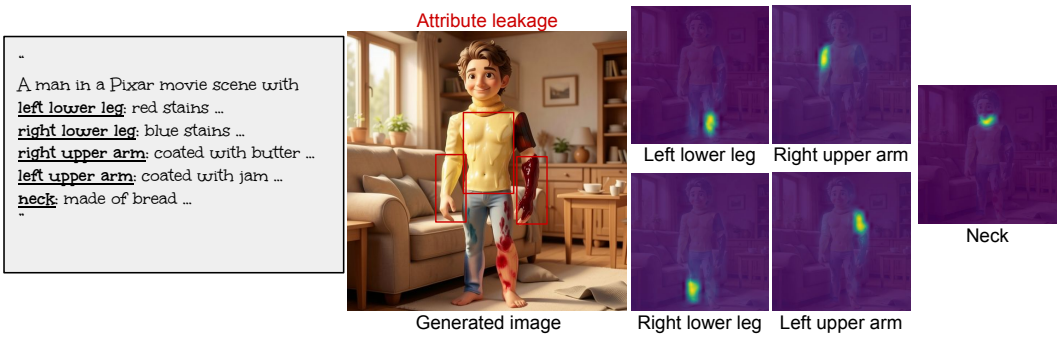

Figure 15: **Example failure case.** Dense text prompts with many parts cause attribute leakage.

D Failure case: dense text prompts w/ many parts

Despite its effectiveness, PARTCONCEPTS has a limitation: scaling beyond four part-attribute pairs occasionally causes attribute leakage, even when spatial localization is accurate. This stems from a distributional shift, as high-density prompts are out-of-distribution (OOD) relative to our training data. An example is shown in fig. 15.

E Compute resources

Our training setup consists of 2 NVIDIA H100 GPUs (96G VRAM each) with one INTEL(R) XEON(R) GOLD 6526Y CPU and 512G memory. Most of our training runs and ablations take roughly 24 hours on this setup.

F Additional qualitative comparisons

We present qualitative comparisons for *fine-grained instruction following* in T2I generation (figs. 17 and 18) and *part instance segmentation* (fig. 19).

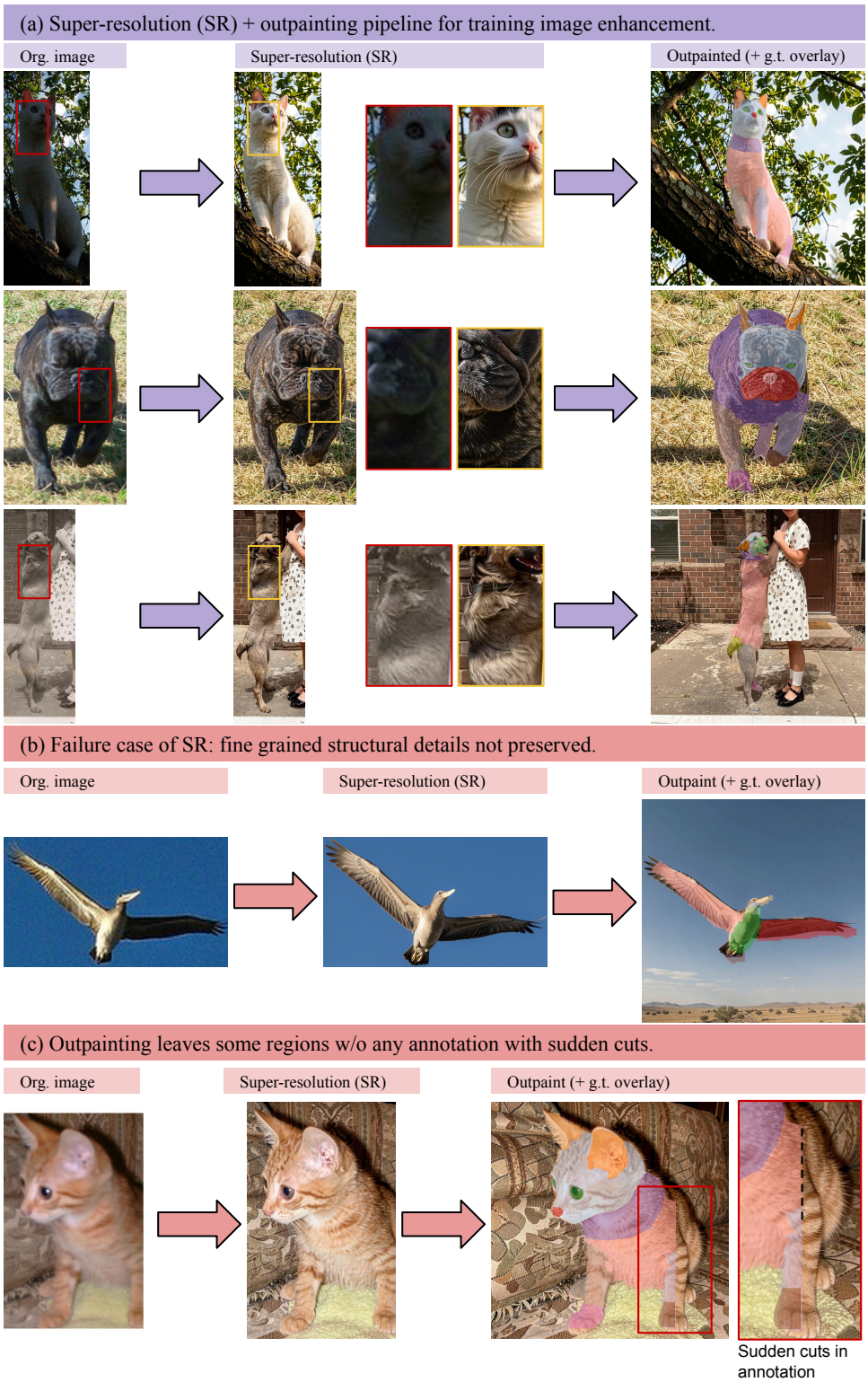

Figure 16: **Super-resolution (SR) + outpainting pipeline for training image enhancement.**

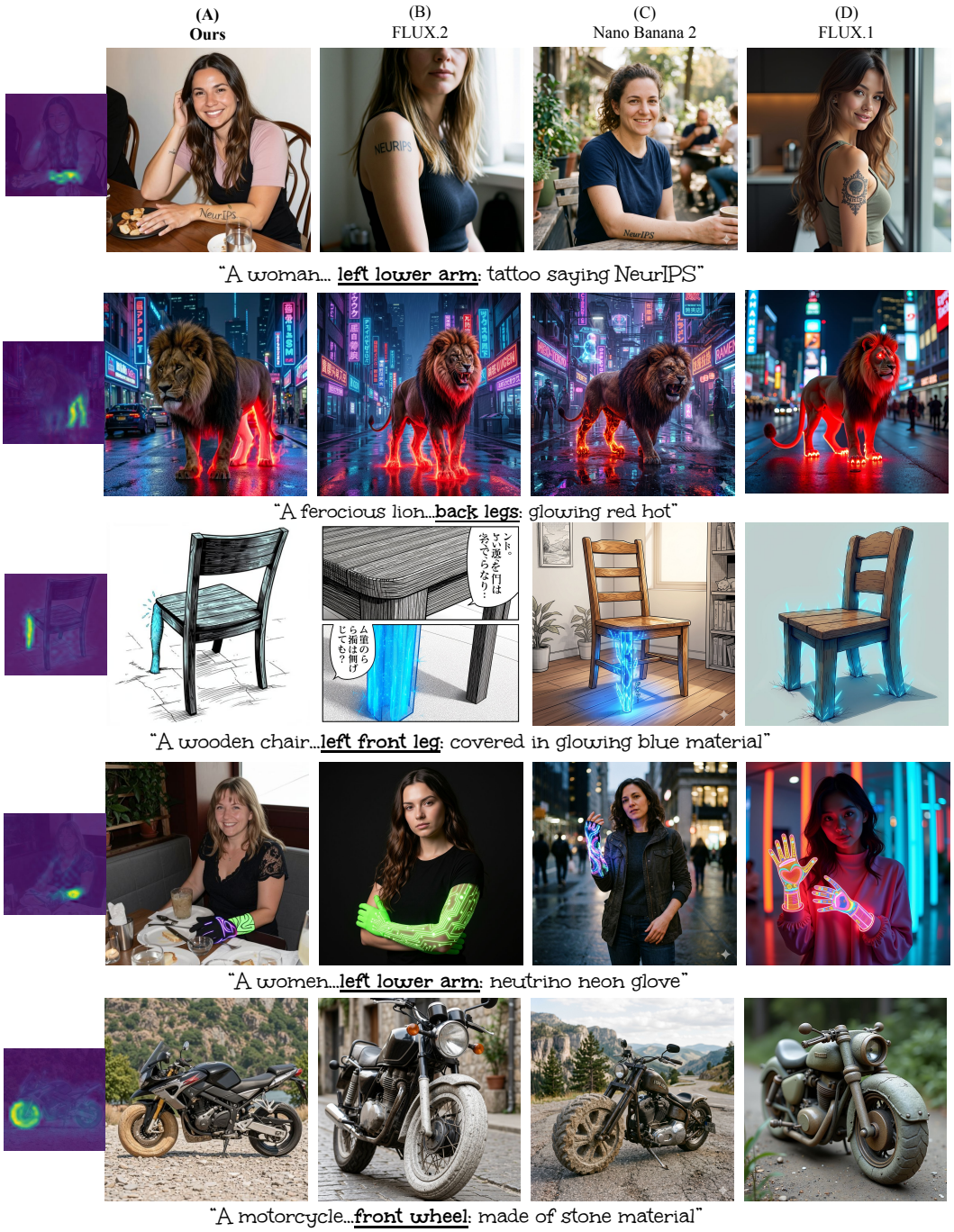

Figure 17: **Qualitative comparisons.** Our method exhibits strong spatial localization, while baselines frequently struggle, failing to disambiguate spatial identifiers in part descriptions (e.g., 'left front leg', 'back legs', 'left lower arm' etc.).



Figure 18: **Qualitative comparisons.** Our method exhibits strong spatial localization, while baselines frequently struggle, failing to disambiguate spatial identifiers in part descriptions (e.g., 'right front leg', 'left handle', 'right arm' etc.).

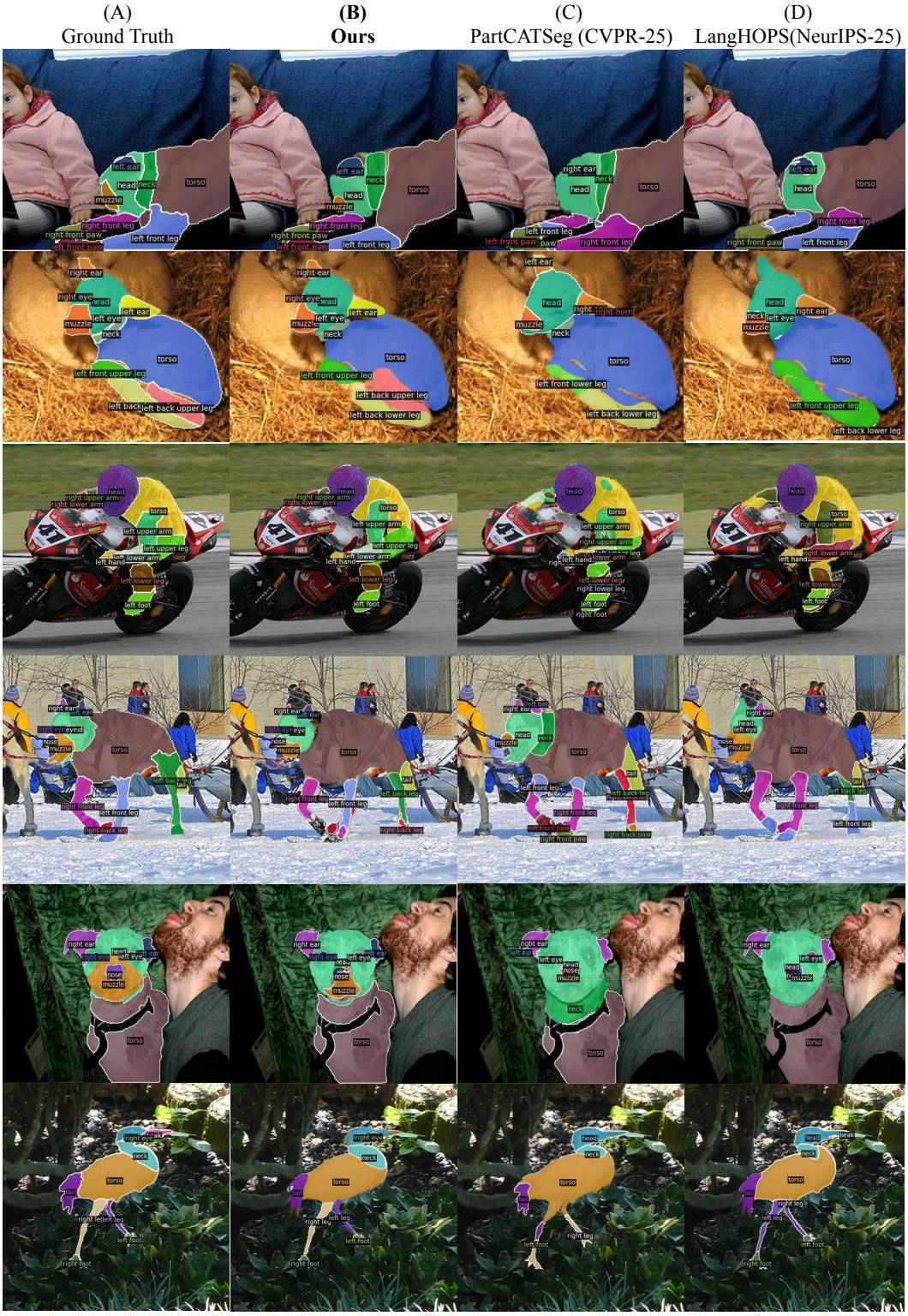

Figure 19: **Qualitative comparison.** Our method outperforms all baselines on various benchmarks, establishing state-of-the-art. Notably, we are significantly better for *part instance segmentation* [15], effectively disambiguating spatially distinct part instances like 'left lower arm' and 'right upper arm'.

G Details about Open-Vocabulary Part Segmentation (OVPS) [14] benchmark

Pascal-Part-116. The Pascal-Part-116 dataset serves as an adaptation of the original Pascal-Part dataset [15], tailored specifically to support Open-Vocabulary Part Segmentation (OVPS) research. It incorporates a combination of base and novel object classes to ensure variety. Notably, the object categories "bird", "car", "dog", "sheep" and "motorbike" are explicitly treated as novel. At the part level, the dataset is structured with 74 base and 42 novel object-specific part categories, establishing a robust framework for evaluating segmentation models. Additional details regarding the dataset are given in table 6.

Table 6: Pascal-Part-116 base, novel categories and their associated part labels.

Object	**Parts**
	Novel Categories (42 Parts)
Bird	beak, eye, foot, head, leg, neck, tail, torso, wing
Car	back, door, front, headlight, license plate, mirror, roof, side, wheel, window
Dog	ear, eye, head, leg, muzzle, neck, nose, paw, tail, torso
Motorbike	handlebar, headlight, saddle, wheel
Sheep	ear, eye, head, horn, leg, muzzle, neck, tail, torso
	Base Categories (74 Parts)
Aeroplane	body, engine, stern, tail, wheel, wing
Bicycle	chainwheel, handlebar, headlight, saddle, wheel
Bottle	body, cap
Bus	back, door, front, headlight, license plate, mirror, roof, side, wheel, window
Cat	ear, eye, head, leg, neck, nose, paw, tail, torso
Cow	ear, eye, head, horn, leg, muzzle, neck, tail, torso
Horse	ear, eye, head, hoof, leg, muzzle, neck, tail, torso
Person	ear, eyebrow, eye, foot, hair, hand, head, leg, lower arm, mouth, neck, nose, torso, upper arm
Pottedplant	plant, pot
Train	back, coach, front, head, headlight, roof, side
Tvmonitor	screen

ADE20K-Part-234. ADE20K-Part-234 is a customized iteration of the ADE20K dataset [23], optimized for OVPS challenges. This dataset provides a highly diverse set of object categories. Similar to Pascal-Part-116, the dataset is divided into base and novel classes. The designated novel object classes encompass "bench", "bus", "fan", "desk", "stool", "truck", "van", "swivel chair", "oven", "ottoman" and "kitchen island". The dataset contains 176 base and 58 novel categories. It introduces unique complexities for segmentation models due to the broad variety of classes and the prevalence of minute object parts that require high-precision mapping. Additional dataset details are provided in table 7.

Table 7: ADE20K-Part-234 base, novel categories and their associated parts.

Object	**Parts**
	Base Categories (176 Parts)

Object	Parts
Airplane	door, fuselage, landing gear, propeller, stabilizer, turbine engine, wing
Armchair	apron, arm, back, back pillow, leg, seat, seat base, seat cushion
Bed	footboard, headboard, leg, side rail
Bookcase	door, drawer, front, side
Cabinet	door, drawer, front, shelf, side, skirt, top
Car	bumper, door, headlight, hood, license plate, logo, mirror, wheel, window, wiper
Chair	apron, arm, back, base, leg, seat, seat cushion, skirt, stretcher
Chandelier	arm, bulb, canopy, chain, cord, highlight, light source, shade
Chest of drawers	apron, door, drawer, front, leg
Clock	face, frame
Coffee table	leg, top
Computer	computer case, keyboard, monitor, mouse
Cooking stove	burner, button panel, door, drawer, oven, stove
Dishwasher	button panel, handle, skirt
Door	door frame, handle, knob, panel
Glass	base, bowl, opening, stem
Lamp	arm, base, canopy, column, cord, highlight, light source, shade, tube
Light	aperture, canopy, diffusor, highlight, light source, shade
Microwave	button panel, door, front, side, top, window
Minibike	license plate, mirror, seat, wheel
Person	arm, back, foot, gaze, hand, head, leg, neck, torso
Pool table	bed, leg, pocket
Refrigerator	button panel, door, drawer, side
Sconce	arm, backplate, highlight, light source, shade
Shelf	door, drawer, front, shelf
Sink	bowl, faucet, pedestal, tap, top
Sofa	arm, back, back pillow, leg, seat base, seat cushion, skirt
Table	apron, drawer, leg, shelf, top, wheel
Television receiver	base, buttons, frame, keys, screen, speaker
Toilet	bowl, cistern, lid
Traffic light	housing, pole
Wardrobe	door, drawer, front, leg, mirror, top
Washer	button panel, door, front, side
Novel Categories (58 Parts)	
Bench	arm, back, leg, seat
Bus	bumper, door, headlight, license plate, logo, mirror, wheel, window, wiper
Desk	apron, door, drawer, leg, shelf, top
Fan	blade, canopy, tube
Kitchen island	door, drawer, front, side, top
Ottoman	back, leg, seat
Oven	button panel, door, drawer, top
Stool	leg, seat
Swivel chair	back, base, seat, wheel
Truck	bumper, door, headlight, license plate, logo, mirror, wheel, window
Van	bumper, door, headlight, license plate, logo, mirror, taillight, wheel, window, wiper

PartImageNet. Derived from ImageNet [57], PartImageNet [27] encompasses roughly 24,000 images representing 158 distinct categories, complete with meticulous part-level annotations. Utilizing WordNet's hierarchical structure, these categories are organized into 11 broader superclasses. Because PartImageNet does not feature pre-established base and novel splits for cross-dataset testing, a zero-shot evaluation protocol is established by isolating 40 representative object categories which are further divided into 25 base and 15 novel object classes [34].

Table 8: PartImageNet base, novel object categories and their associated parts

Base Object Categories (25)	Novel Object Categories (15)	Part Classes
Tiger, Giant panda, Leopard, Gazelle	Ice bear, Impala, Golden retriever	Head, Body, Foot, Tail
Green mamba	Indian cobra	Head, Body
Green lizard, Komodo dragon, Tree frog	Box turtle, American alligator	Head, Body, Foot, Tail
Yawl, Pirate	Schooner	Body, Sail
Barracouta, Goldfish, Killer whale	Tench	Head, Body, Fin, Tail
Albatross, Goose	Bald eagle	Head, Body, Wing, Foot, Tail
Garbage truck, Minibus, Ambulance	Jeep, School bus	Body, Tire, Side Mirror
Mountain bike, Moped	Motor scooter	Body, Head, Seat, Tire
Gorilla, Orangutan	Chimpanzee	Head, Body, Hand, Foot, Tail
Beer bottle, Water bottle	Wine bottle	Mouth, Body
Warplane	Airliner	Head, Body, Engine, Wing, Tail

PartImageNet OOD PartImageNet [27] offers an alternative split designed for few-shot learning, which ensures non-overlapping classes across the training and validation sets. This few-shot split includes 109 base object classes in the training set and 19 novel object classes in the validation set. Detailed information about the base and novel classes along with their corresponding superclasses, is presented in table 9.

Table 9: PartImageNet (OOD) base, novel object categories and their associated parts.

Superclass	Base Object Categories (109)	Novel Object Categories (19)
Bird	albatross, spoonbill, black stork, dowitcher, American egret, goose, ruddy turnstone, bee eater, kite	little blue heron, bald eagle
Quadruped	impala, Egyptian cat, warthog, otter, Tibetan terrier, timber wolf, polecat, water buffalo, ox, redbone, English springer, tiger, American black bear, leopard, hartebeest, vizsla, Brittany spaniel, giant panda, Boston bull, ram, cairn, Arabian camel, fox squirrel, Eskimo dog, Irish water spaniel, Saluki, Walker hound, cheetah, gazelle, soft-coated wheaten terrier, bighorn, brown bear, chow, weasel	golden retriever, cougar, ice bear, mink, Saint Bernard
Snake	night snake, boa constrictor, green mamba, thunder snake, green snake, hognose snake, sidewinder, horned viper, diamondback, rock python, garter snake, vine snake	Indian cobra

Continued on next page...

Superclass	Base Object Categories (109)	Novel Object Categories (19)
Reptile	Gila monster, common newt, green lizard, bullfrog, American alligator, leatherback turtle, spotted salamander, box turtle, tailed frog, African chameleon, Komodo dragon, agama, frilled lizard, loggerhead	whiptail, alligator lizard
Boat	yawl, pirate	trimaran
Fish	goldfish, coho, tench, anemone fish, killer whale	barracouta, great white shark
Car	garbage truck, minibus, ambulance, snowplow, golfcart, police van, minivan, convertible, limousine, recreational vehicle, go-kart, tractor, school bus, racer	beach wagon, cab
Bicycle	motor scooter, tricycle, mountain bike	unicycle
Biped	gorilla, gibbon, guenon, macaque, patas, howler monkey, chimpanzee, proboscis monkey, spider monkey, baboon, colobus, capuchin	siamang, marmoset
Bottle	beer bottle, pop bottle, pill bottle	water bottle
Aeroplane	airliner	

H The Pascal-Part (org.) benchmark for fine-grained part instance segmentation

The original Pascal-Part [15] consists of fine-grained part labels, with spatial identification of part instances (e.g., left lower arm, right upper arm). However, the Pascal-Part-116 (PPS-116) [14] split in the OVPS suite collapses these spatially distinct instances, e.g., 'left ear' and 'right ear', into a single 'ear' instance (see fig. 20). This removes precision necessary for various real-world applications in robotics and T2I generation, reducing the problem from *part instance segmentation* to *part semantic segmentation*. Therefore, we additionally evaluate our approach on the original Pascal-Part [15] dataset in section 4.2, where our method significantly outperforms all baselines.



Figure 20: **OVPS [14] vs. original Pascal-Part [15].** The OVPS Pascal-Part-116 [14] benchmark merges spatially distinct part instances from the original Pascal-Part [15]. For example, 'left ear' and 'right ear' are merged into a single 'ear' instance.

I Part-Level IoU scores

We present the part-level IoU scores for our model on the original Pascal-Part [15] benchmark in table 10.

Table 10: Part-Level IoU scores for our model on original Pascal-Part [15] benchmark.

Part Name	IoU (%)	Part Name	IoU (%)
Bird beak	38.31	Cow right front lower leg	65.10
Bird head	66.78	Cow right front upper leg	53.92
Bird left eye	44.32	Cow right horn	59.67
Bird left foot	45.70	Cow tail	64.80
Bird left leg	18.46	Cow torso	90.10
Bird left wing	23.73	Dog head	66.45
Bird neck	47.95	Dog left back leg	50.91
Bird right eye	33.15	Dog left back paw	66.36
Bird right foot	53.22	Dog left ear	67.84
Bird right leg	23.58	Dog left eye	58.97
Bird right wing	23.99	Dog left front leg	58.01
Bird tail	52.86	Dog left front paw	67.58
Bird torso	77.04	Dog muzzle	37.85
Bicycle back wheel	16.96	Dog neck	51.89
Bicycle chain wheel	72.35	Dog nose	14.89
Bicycle front wheel	25.25	Dog right back leg	48.70
Bicycle handlebar	90.34	Dog right back paw	56.96
Bicycle saddle	28.48	Dog right ear	65.76
Bottle body	98.37	Dog right eye	52.71
Bottle cap	74.14	Dog right front leg	52.89
Bus back license plate	0.00	Dog right front paw	70.37
Bus back side	86.44	Dog tail	76.54
Bus door	5.12	Dog torso	82.59
Bus front license plate	0.05	Motorbike back wheel	9.30
Bus front side	79.51	Motorbike front wheel	30.40
Bus headlight	44.13	Motorbike handlebar	100.00
Bus left mirror	47.61	Motorbike headlight	98.20
Bus left side	78.63	Pottedplant plant	89.92
Bus right mirror	22.92	Aeroplane body	85.10
Bus right side	71.79	Aeroplane engine	32.73
Bus roof side	23.51	Aeroplane left wing	58.83
Bus wheel	67.91	Aeroplane right wing	55.07
Bus window	81.83	Aeroplane stern	64.90
Car back license plate	0.05	Aeroplane tail	0.01
Car back side	72.68	Aeroplane wheel	84.53
Car door	4.09	Horse head	76.62
Car front license plate	0.01	Horse left back hoof	48.27
Car front side	68.26	Horse left back lower leg	59.84
Car headlight	40.81	Horse left back upper leg	56.14
Car left mirror	40.84	Horse left ear	76.89
Car left side	62.36	Horse left eye	42.65
Car right mirror	53.83	Horse left front hoof	44.96
Car right side	65.18	Horse left front lower leg	48.89
Car roof side	25.36	Horse left front upper leg	52.91
Car wheel	85.23	Horse muzzle	81.33
Car window	73.71	Horse neck	74.18
Cat head	84.15	Horse right back hoof	51.54
Cat left back leg	34.10	Horse right back lower leg	63.90
Cat left back paw	55.55	Horse right back upper leg	59.06

Continued on next page...

Part Name	IoU (%)	Part Name	IoU (%)
Cat left ear	75.67	Horse right ear	72.05
Cat left eye	57.15	Horse right eye	51.90
Cat left front leg	47.68	Horse right front hoof	46.46
Cat left front paw	55.64	Horse right front lower leg	51.18
Cat neck	39.08	Horse right front upper leg	52.58
Cat nose	44.14	Horse tail	78.49
Cat right back leg	51.33	Horse torso	88.22
Cat right back paw	63.84	Person hair	9.78
Cat right ear	73.88	Person head	55.46
Cat right eye	52.60	Person left ear	46.82
Cat right front leg	46.43	Person left eye	34.89
Cat right front paw	55.07	Person left eyebrow	25.47
Cat tail	79.65	Person left foot	74.31
Cat torso	83.85	Person left hand	71.61
Cow head	70.53	Person left lower arm	69.44
Cow left back lower leg	59.36	Person left lower leg	71.17
Cow left back upper leg	38.47	Person left upper arm	77.45
Cow left ear	70.87	Person left upper leg	72.54
Cow left eye	54.49	Person mouth	50.51
Cow left front lower leg	70.13	Person neck	61.79
Cow left front upper leg	60.87	Person nose	56.53
Cow left horn	74.17	Person right ear	57.70
Cow muzzle	72.74	Person right eye	36.92
Cow neck	48.46	Person right eyebrow	24.60
Cow right back lower leg	66.76	Person right foot	69.60
Cow right back upper leg	46.54	Person right hand	76.79
Cow right ear	80.47	Person right lower arm	70.89
Cow right eye	60.09	Person right lower leg	70.73
Sheep head	61.07	Person right upper arm	75.25
Sheep left back lower leg	63.56	Person right upper leg	71.44
Sheep left back upper leg	39.47	Person torso	86.38
Sheep left ear	58.56	Pottedplant pot	86.96
Sheep left eye	32.48	Train coach back side	6.91
Sheep left front lower leg	67.52	Train coach front side	0.10
Sheep left front upper leg	59.81	Train coach left side	0.51
Sheep left horn	42.43	Train coach	58.68
Sheep muzzle	62.75	Train coach right side	27.52
Sheep neck	51.07	Train coach roof side	33.68
Sheep right back lower leg	64.42	Train head	5.09
Sheep right back upper leg	49.52	Train head back side	19.04
Sheep right ear	49.50	Train head front side	60.61
Sheep right eye	26.24	Train head left side	53.95
Sheep right front lower leg	64.84	Train head right side	62.64
Sheep right front upper leg	61.21	Train head roof side	57.36
Sheep right horn	46.81	Train headlight	1.55
Sheep tail	51.58	TV monitor screen	100.00
Sheep torso	90.67	-	-

Continued on next page...

J User study

The user study was conducted to evaluate attribute localization performance of our method against other baselines. The study was an A/B test where participants were shown two images, one generated

by our method and the other by a baseline method. They were asked to identify which image was 'better' w.r.t. our evaluation metric (attribute localization).

To explain the task to the participants, we provided four illustrative example questions, shown in fig. 21.

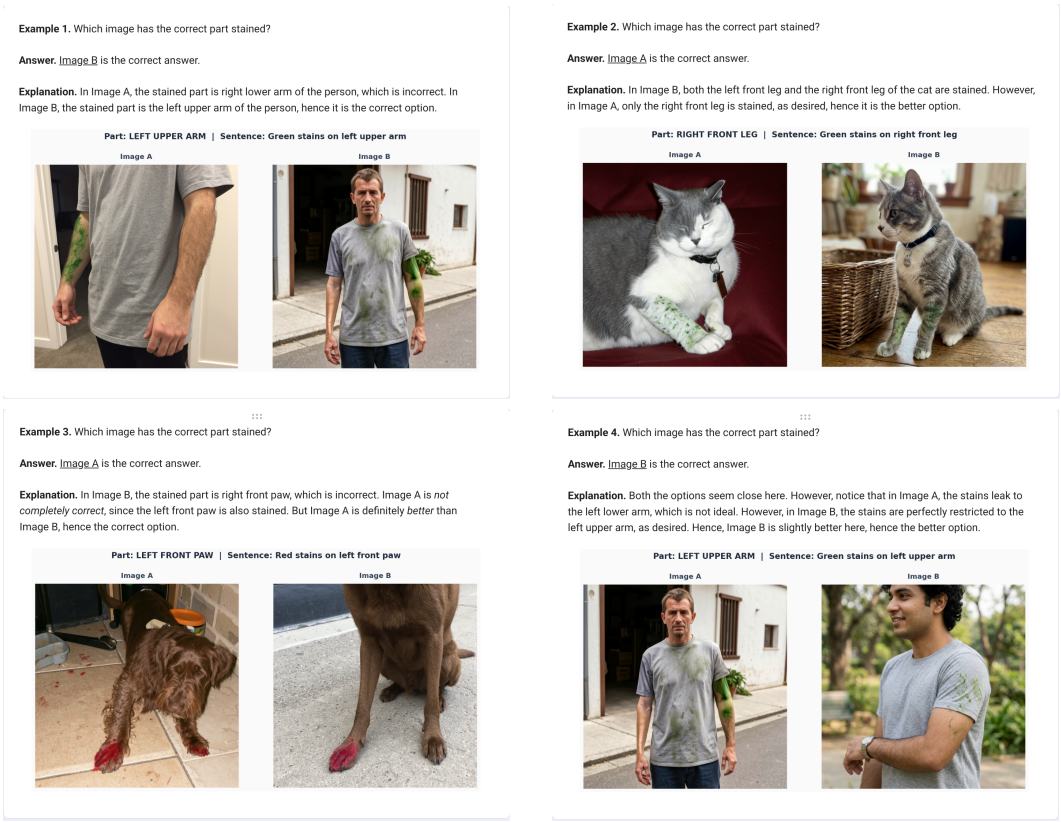

Figure 21: Examples of questions shown to participants in our user study.

The study consisted of 51 questions, 17 questions per baseline method. We collected responses from 40 participants, resulting in a total of 2040 responses. The results of the user study are presented in fig. 6.

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: In our abstract, we claim to improve text-aligned **part-level** understanding of a T2I model while preserving its prior. We validate this claim by showing that the improved part localization can be used for *part-level segmentation* (sections 4.1 and 4.2). We also show that this strong localization enables precise part-level attribute binding, thus providing *fine-grained instruction following* (section 4.3) in T2I generation.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 are discussed in section 5 of the main paper.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] 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 describe the details of our method in section 3. We also describe implementation and training details in section 4.1 and section 4.3. Together these ensure reproducibility of our results.

Guidelines:

- The answer [N/A] 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: [No]

   Justification: We will release both data and code post acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All details regarding data splits can be found in Appendix G. All training and implementation details are mentioned in section 4.1 and section 4.3.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: Error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer [N/A] 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: Our compute requirements are mentioned in Appendix E

   Guidelines:

   - The answer [N/A] 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: The paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [No]

    Justification: Our work focuses on the fundamental technical aspects of part-level understanding in Text-to-Image (T2I) models. While we acknowledge the broad societal implications of T2I technology, this research is a controlled diagnostic study that does not introduce new risks/impact. Consequently, a standalone discussion of societal impact was omitted in favor of focusing on the technical analysis.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our model inherits safeguards from the base model [1] upon which our model has been trained.

    Guidelines:

    - The answer [N/A] 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: The models and data used in the paper are properly credited, the license and terms of use are correctly respected.

    Guidelines:

    - The answer [N/A] 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: Documentation for the assets is already provided throughout the paper's content. The assets will be made publicly available post acceptance.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: We conducted a user study to evaluate generation accuracy. Participation was entirely voluntary, no compensation was given. A screenshot of the user study has been added in Appendix J

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The paper involves a user study for evaluating generation accuracy; this does not involve any risks to the participants. We have received the necessary IRB approvals for this.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: A dataset preparation stage requires usage of VLMs for captioning images, see Appendix C. We ensure that our process adheres to the NeurIPS code.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.