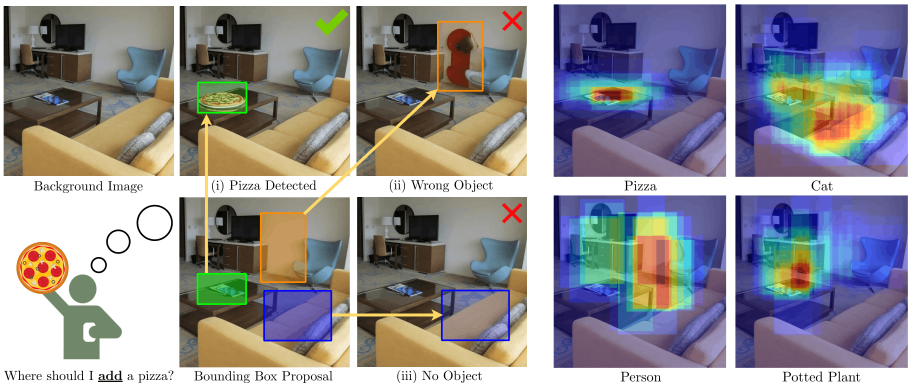
arXiv:2604.10675v1 [cs.CV] 12 Apr 2026

HiddenObjects: Scalable Diffusion-Distilled Spatial Priors for Object Placement

Marco Schouten[1,3] Ioannis Siglidis[2,3] Serge Belongie[2,3]
Dim P. Papadopoulos[1,3]

[1]Technical University of Denmark [2]University of Copenhagen [3]Pioneer Centre for AI
https://hidden-objects.github.io/



(a) Spatial Prior Extraction (b) Multi-class Spatial Priors

Fig. 1: Spatial Priors for Object Placements. (a) We score candidate bounding boxes for object insertion using a diffusion-based inpainting pipeline. In this example, three pizza locations are considered. First, we attempt to insert the object in the designated locations via inpainting. Then, we evaluate these locations, yielding three possible outcomes: (i) "Pizza Detected", (ii) "Wrong Object" or (iii) "No Object". Only plausible placements (i.e., (i)) contribute to the spatial prior. (b) Aggregating predictions across many scenes and object categories produces heatmaps that capture human-aligned placement expectations (e.g., pizza on the table, cat on the sofa).

Abstract. We propose a method to learn explicit, class-conditioned spatial priors for object placement in natural scenes by distilling the implicit placement knowledge encoded in text-conditioned diffusion models. Prior work relies either on manually annotated data, which is inherently limited in scale, or on inpainting-based object-removal pipelines, whose artifacts promote shortcut learning. To address these limitations, we introduce a fully automated and scalable framework that evaluates dense object placements on high-quality real backgrounds using a diffusion-based inpainting pipeline. With this pipeline, we construct HiddenObjects, a large-scale dataset comprising 27M placement annotations, evaluated across 27k distinct scenes, with ranked bounding box insertions for different images and object categories. Experimental results show that our spatial priors outperform sparse human annotations on a downstream

image editing task (3.90 vs. 2.68 VLM-Judge), and significantly surpass existing placement baselines and zero-shot Vision-Language Models for object placement. Furthermore, we distill these priors into a lightweight model for fast practical inference (230,000× faster).

Keywords: Object Placement · Spatial Prior · Diffusion Models

1 Introduction

Should a pizza ever appear on the floor? As products of human priors, natural scene images reflect strong spatial regularities: pizzas rest on tables, books lie on shelves, and clocks hang on walls. These contextual spatial priors have long been implicitly exploited to improve visual recognition and scene understanding [12, 15, 41, 49]. While these priors underly most visual datasets, they have not been explicitly modeled or learned at scale. Explicitly modeling these spatial priors enables applications, such as realistic image editing [8,45] or outlier detection [6].

Recent work on object placement has approached the extraction of spatial priors either by removing existing objects from images via inpainting [35, 52, 53] or by manually annotating exemplar object placements for new objects [26, 31]. While the first approach can produce priors automatically, the removed objects can contain both subtle and strong background artifacts (see Fig. 2) that facilitate shortcut learning. The second approach avoids background artifacts, but as manual annotation is not scalable, it becomes a trade-off between low-dataset diversity (as not a lot of images can be annotated) and unconfident annotations (as images can only be annotated by few people). Meanwhile, in both cases, datasets contain sparse object placements (i.e., one location per instance) and often no information related to the likelihood of these proposals.

Instead, our main observation is that the notion of spatial priors is closely related to image completion: given a scene image, predicting plausible object placements is essentially a matter of filling in missing content. Early non-parametric approaches showed that plausible completions can be retrieved from the context of millions of images [19]. Interestingly, even early GAN-based image editing tools revealed that generative models implicitly learn contextual constraints. They understand where objects belong and resist placing them in implausible regions (e.g., refusing to draw doors in the sky) [4, 5, 34]. More recently, text-conditioned diffusion models have shown remarkable abilities in image generation and inpainting [42, 44, 61] enabled by training on billions of images [47]. Unlike previous approaches, diffusion models offer fine-grained controllability, allowing users to add or blend objects across all plausible locations of an input scene.

Motivated by their abilities, we propose a framework to extract scalable, dense, and explicit spatial priors directly from diffusion models without **any** human intervention. Practically, our key observation is that while diffusion models can inpaint objects in semantically plausible locations (Fig. 1.i), they can also produce wrong inpaintings (Fig. 1.ii) or refuse inpainting all together (Fig. 1.iii). Using state-of-the-art object detectors [43] we can verify the validity of an insertion. Then, using an image-based reward model [56], we can rank preferences

Fig. 2: Background artifacts in PIPE [52]. Object removal pipelines are effective at removing the requested object but leave visible traces that can facilitate shortcut learning, e.g., (left) blurry area and the shadow presence, or (right) larger artifacts.

across verified placements. Exhaustive testing of multiple insertions using a sliding window can densely capture and rank all plausible object locations within a scene, finally producing explicit, class-conditioned spatial priors (see Fig. 1b).

Using our pipeline, we construct a large-scale dataset of 27M object placement annotations in 27K scenes from Places365 [63]. Moreover, we distill these priors into a transformer-based model that predicts bounding boxes for new scenes efficiently, enabling fast inference for downstream applications. Our experimental results reveal three key findings. First, on the downstream task of object insertion, our automatically generated annotations yield significantly higher image editing quality than human annotations (3.90 vs. 2.68 average ImgEdit-Judge score [57]), demonstrating that manual placement labels are inherently ambiguous and less suitable for inpainting. Second, the extracted spatial priors closely align with human placement expectations while exhibiting greater spatial diversity than standard object-centric recognition datasets, which suffer from severe center bias. Third, our distilled model outperforms all existing placement baselines and zero-shot VLMs by a large margin, confirming that dense, reward-weighted spatial priors are essential for learning robust placement distributions.

Our contributions are threefold: **(1)** We present a scalable framework to extract multi-class spatial priors from diffusion models for input images. **(2)** We construct HiddenObjects, a dataset of 27M densely evaluated object placements in 27K images. **(3)** We distill these priors into a lightweight transformer model that achieves a 230,000× speedup and 300× memory reduction over the full exhaustive extraction pipeline, enabling fast object placement proposals for downstream applications.

2 Related Work

Spatial Priors. We define a *spatial prior* as the probability distribution of valid bounding box locations for inserting a foreground object in a given scene. Recent approaches learn spatial priors through VLMs that reason over geometric structure [1, 22]. Others model the spatial prior for domain-specific objects [36, 37].

Table 1: Object Placement Datasets. Comparison of common datasets for learning object placement models. Manually annotated datasets are limited in scale, while automated approaches often introduce visual artifacts via inpainting. Our proposed HiddenObjects dataset achieves both scalability, high-quality dense bounding box annotations, and artifact-free backgrounds.

Dataset	# Annotation Boxes	# Positive Box (per instance)	# Negative Box (per instance)	Real Images	Dense Priors	Scalable	Artifact-free Background
DreamEditBench [26]	220	1	0	✓	✗	✗	✓
MureCom [32]	640	1	0	✓	✗	✗	✓
OPA [31]	0.07M	1.9	3.8	✓	✗	✗	✓
OPA-ext [31]	0.15M	2.2	4.3	✓	✗	✗	✓
PIPE [52]	0.89M	1	0	✗	✗	✓	✗
SAM-FB [20]	3M	1	0	✗	✗	✓	✗
HiddenObjects	**27M**	**77.7**	**926.3**	✓	✓	✓	✓

However, such methods are domain-constrained, dependent on 3D annotations, and do not generalize to scene images. Instead, we learn spatial priors on scene images by distilling the implicit knowledge about object location priors that is embedded in diffusion models from their internet scale image training data [47].

Object placement aims to identify suitable locations for inserting a foreground object into a background scene with the highest semantic and geometric consistency. Early *context-based methods* [13, 51] are limited to evaluating placement for input locations, lacking the ability to propose new ones. More recently, *location models* learned placement prediction distributions from manually annotated data [10, 24, 28, 33, 39, 59, 62, 65, 67, 68]. Yet, manually annotated data collection is not scalable, therefore object placement models trained on these data struggle to generalize. Furthermore, other datasets that provide annotated locations, copy-paste foreground objects into backgrounds scenes, lacking perspective adaptation [31, 40]. To resolve this, a set of synthetic approaches [16, 52, 53, 66] train location models from large-scale synthetic data obtained by removing foreground objects from background scenes via inpainting. Different to our approach, the construction of these synthetic datasets often introduces inpainting artifacts (see Fig. 2) in the original object region of *input background images*, which facilitates shortcut learning when used to train object placement models.

Vision Language Models (VLMs) are an alternative paradigm to inferring object placement locations. Some methods [7, 20, 58] use VLM-generated context descriptions to support outpainting of background regions around a given foreground object to create annotated pairs of object insertion, while others [25, 27] rely on VLMs to predict placement locations directly. Yet, VLM-generated context descriptions lack the diversity captured in large scene datasets, such as Places365 [63], and often unreliably predict locations directly. Closer to us, some methods [17, 23, 27, 35, 48] rely on diffusion models implicit spatial priors for predicting placement locations. However, these predictions are tied to their downstream image editing task, limiting their use in other models or applications. Unlike prior work, our methodology enables fully automated, artifact-free annotation of object placements at scale by distilling spatial object priors from

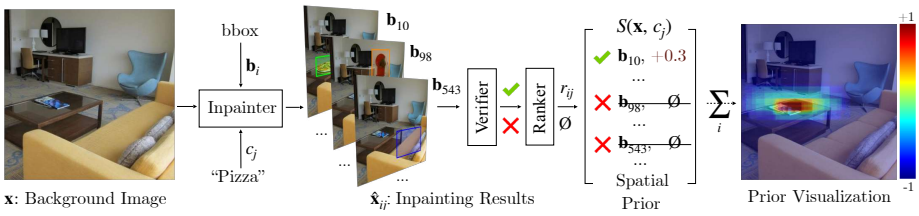


Fig. 3: Spatial Prior Extraction. Given a background image $\mathbf{x}$, the inpainter $\mathcal{I}$ synthesizes an image $\hat{\mathbf{x}}_{ij}$ for target class c_j at a bounding box $\mathbf{b}_i$. The verifier $\mathcal{V}$ evaluates the inpainted images $\hat{\mathbf{x}}_{ij}$, yielding a detected bounding box and a human-aligned preference score r_{ij}. Repeating this process across multiple sampled boxes yields the ground-truth spatial prior $S(x, c_j)$ for the j-th class. The heatmap visualizes the aggregated dense distribution of preferences over spatial locations (no, mid, good).

text-conditioned diffusion models. This allows modeling the full distribution of plausible and implausible object placements in a realistic distribution of scene images, useful in a variety of downstream tasks.

Datasets. Existing datasets for object placement typically require a background image without the target object and a bounding box that annotates a placement location (Tab. 1). There are two main strategies for constructing such datasets. The first, relies on manual annotation of real images, where annotators directly mark potential regions for given objects in real scenes as positive or negative [26, 31, 32]. The second, utilizes object-removal pipelines to obtain the object-free background from real scenes that contain the target object in a valid location [35, 52, 53]. Manual annotations are not scalable, and object removal pipelines yield artifacts in the *input background image*. In contrast, our pipelines provide automatically generated bounding box annotations for real images.

3 HiddenObjects Dataset

The main contribution of this work is a generative automatic pipeline to extract *spatial priors* for object placement on real background images. In Sec. 3.1, we describe our framework, and, in Sec. 3.2, we discuss our dataset construction.

3.1 Spatial Prior Extraction Framework

The construction of a spatial prior $\mathbf{S}(\mathbf{x}, c)$ between a background image x and an object c can be broken down into three fundamental operations outlined in Fig. 3. First, an inpainter $\mathcal{I}$ synthesizes class-consistent object insertions into a set of proposed regions. Secondly, a verifier $\mathcal{V}$ validates whether the inpainter correctly added the target object in the proposed region. Third, for each verified insertion, a Ranker $\mathcal{R}$ quantifies preference conditional to the object.

Spatial Prior. In the context of our dataset, a spatial prior $\mathbf{S}(\mathbf{x}, c)$ for a given input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ and a category $c \in \mathcal{C}$ is defined as a set:

$$\mathbf{S}(\mathbf{x}, c) \equiv \{(\mathbf{b}_i, r_i), i \in [0, N-1]\} \tag{1}$$

where $r_i \in \mathbb{R}$ is the score of a corresponding bounding box $\mathbf{b}_i = (x_i, y_i, w_i, h_i)$, with box center (x_i, y_i) and box size (w_i, h_i) for a fixed set of N boxes. We define a deterministic set of boxes $\mathbf{b_i}$ produced by a sliding window of different resolutions which evenly span the whole image (more details in Appendix E). For simplicity, we assume all images to be in the same resolution.

Inpainter. The goal of the inpainter $\mathcal{I}$ is to produce a insertion proposal at different locations in the input image. We denote the inpainting result:

$$\hat{\mathbf{x}}_{ij} \equiv \mathcal{I}(\mathbf{x}, \mathbf{b}_i, c_j), \tag{2}$$

in the region defined by the bounding box $\mathbf{b}_i$ for class c_j. Note, that it is possible that $\exists\, i, j : \hat{\mathbf{x}}_{ij} \approx \mathbf{x}$ or that the inpainting doesn't contain c_j (see Fig. 1 **(a)**).

Verifier. The goal of the verifier $\mathcal{V}$ is to identify whether the target category has been correctly inpainted in the image after the inpainting stage. Concretely, given a synthesized image $\hat{\mathbf{x}}_{ij}$ for an input bounding box $\mathbf{b}_i$ and target class c_j, we use a conditional verifier $\mathcal{V}(\mathbf{x}, c_j)$ that produces a set of $\mathbf{B}_{ij}$ detections:

$$\mathbf{B}_{ij} \equiv \{\mathbf{b} \mid (\mathbf{b}, \psi) \in \mathcal{V}(\hat{\mathbf{x}}_{ij}, c_j),\ \psi \geq \tau\} \tag{3}$$

where each $\mathbf{b}$ is predicted with confidence $\psi \in [0, 1]$ and $\tau \in [0, 1]$ is the confidence threshold. When $|\mathbf{B}_{ij}| = 0$ we consider the prediction to be negative by setting $\mathbf{b}_{ij} = \varnothing$. Otherwise, we select the verifier bounding box $\mathbf{b}$ with the highest **I**ntersection **o**ver **U**nion (IoU) to the input bounding box $\mathbf{b}_i$:

$$\mathbf{b}_{ij} = \arg\max_{\mathbf{b} \in \mathbf{B}_{ij}} \text{IoU}(\mathbf{b}, \mathbf{b}_i) \tag{4}$$

Instead of adding objects, inpainting may replace objects that already exist in the background image $\mathbf{x}$. To guarantee that no object placement corresponds to pre-existing objects of c_j, we set $b_{ij} = \varnothing$ if the verified bounding box $\mathbf{b}_{ij}$ has IoU ≥ 0.5 to verified objects of the same class in $\mathbf{x}$.

Ranker. After verifying the correct insertions, a conditional ranker $\mathcal{R}(x, c)$ ranks which of those insertions are preferred against others. This is an essential step in constructing the final object placement prior as it provides a way of quantifying preference across different placement locations. Namely, for each image $\hat{\mathbf{x}}_{ij}$ and class c_j the final extracted prior $\mathbf{S}(\mathbf{x}, c_j)$ can be computed as:

$$\mathbf{S}(\mathbf{x}, c_j) = \{(\mathbf{b}_{ij}, r_{ij}),\ \forall\, \mathbf{b}_{ij} \neq \varnothing\},\ \text{where } r_{ij} \equiv \mathcal{R}(\hat{\mathbf{x}}_{ij}, c_j). \tag{5}$$

3.2 Dataset Construction

To construct HiddenObjects, we select images from Places365 [63] for its diversity of background scenes. We randomly select 126 unique background classes, such as promenade, harbor, or classroom. We then pick 10 macro-categories of objects, such as food, furniture, or animal, and for each select 10 classes from COCO [29], resulting in 50 distinct object categories c_j. To avoid implausible insertions, we manually pair object categories with background categories (for object-background selection details see Appendix F). Given a set of valid

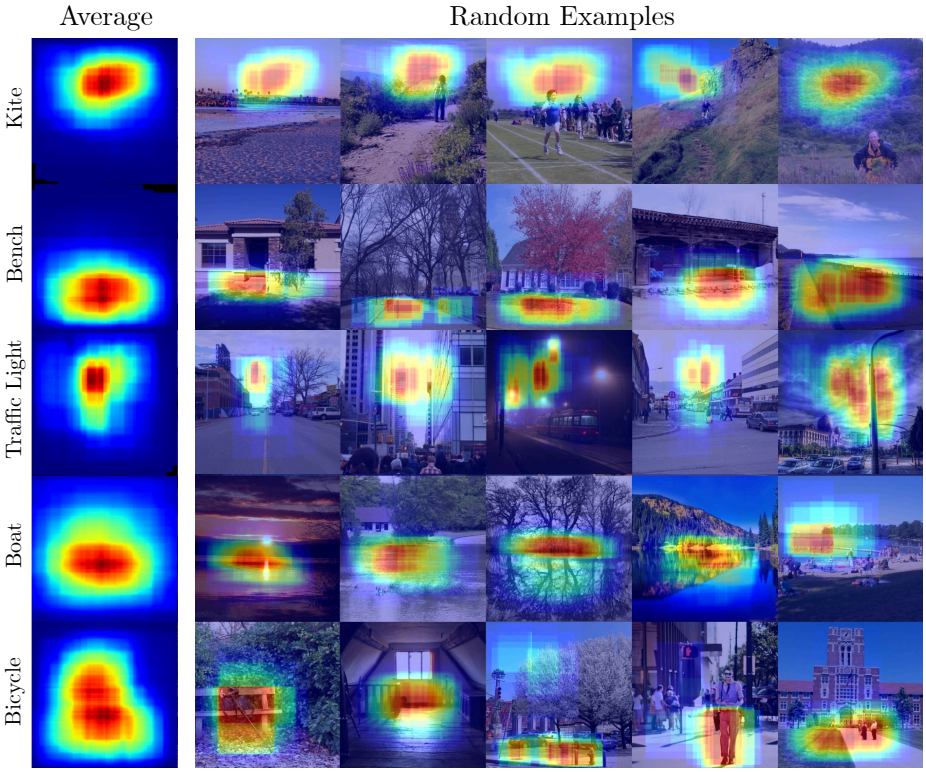


Fig. 4: Spatial Prior (HiddenObjects) aggregated as a heatmap. (**Left**) We visualize the mean aggregated spatial prior across all images of HiddenObjects for different classes listed on the left. Classes such as *bench* and *boat* tend to appear near the bottom of images, while *kite* occurs more frequently near the top. These patterns reflect both semantic regularities (e.g., kite in the sky) and inherent photographic biases in image datasets. (**Right**) We visualize per-instance random example of spatial priors.

background and foreground object pairs, we randomly sample their combinations until we compute a total of 30K spatial priors, with $1,004$ bounding boxes for each. After the ranker stage, the final dataset consists of 27,010 positively annotated backgrounds and 2,097,587 positive bounding box annotations. HiddenObjects is split between train (85%), validation (10%) and test (5%) sets.

Framework Implementation Unless stated otherwise, our inpainter-verifier pipeline uses a Qwen-Image model [54] with a ControlNet inpainting adaptor[1] (Qwen+CN) as the inpainter backbone using 20 inference steps and true classifier-free guidance scale set to 4.0. For the verifier module, we utilize Grounded-SAM-2 [43] for open-set object detection with a confidence threshold $\tau = 0.4$ and ImageReward [56] as a preference ranker. A comprehensive ablation justifying these specific component selections is provided in Sec. 4.2.

[1] `https://huggingface.co/InstantX/Qwen-Image-ControlNet-Inpainting`

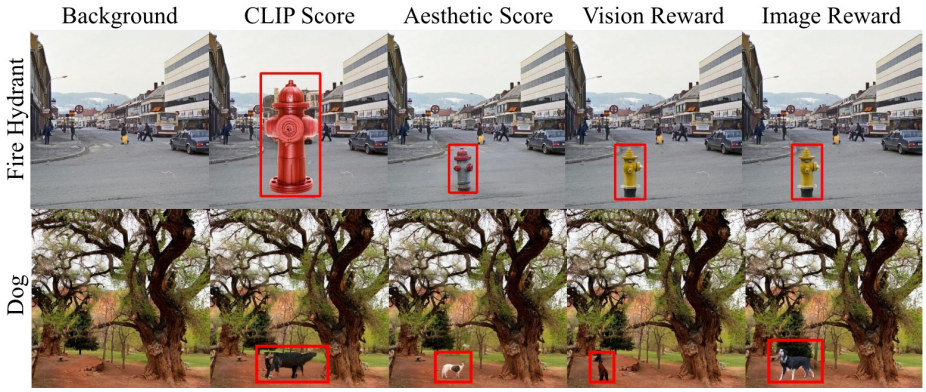


Fig. 5: Verifier Comparison. For each verifier, we display the inpainting generation of the highest-scoring location. CLIP Score often prefers insertions irrespective of global scene context. In contrast, Image and Vision Reward outperform aesthetic score in judging realism and semantic coherence, leading to more human-aligned spatial priors.

Speedup. For the majority of insertion proposals, we expect that the inpainting model will produce inpaintings with missing or wrong objects. Given that the diffusion inpainter is slow, detecting those faulty inpaintings early during the denoising procedure would help avoid unnecessary computational costs. We find that we can do so by measuring how much the noise predicted using the learned denoising network ε_θ varies between the conditional $\varepsilon_\theta(x_t, c_j)$ and the unconditional branch $\varepsilon_\theta(x_t, \varnothing)$ at each denoising step x_t. Namely, we find that the L1-difference between the two branches correlates highly with prompt-adherence, where successful generations exhibit much higher values than failed generations. More specifically, we compute a threshold where we can detect faulty boxes with a total recall of 81% for a single denoising step, leaving only 30% of proposals for inpainting, yielding a 2.4x speedup without degrading prior quality. Our detailed analysis can be found in Appendix D.

4 Pipeline Validation and Dataset Analysis

We evaluate our diffusion-based spatial prior extraction pipeline along three axes: Sec. 4.1 key component selection, Sec. 4.2 downstream image editing quality and human alignment, and Sec. 4.3 dataset bias analysis. We report both qualitative analyses and quantitative metrics assessing realism and robustness.

4.1 Key Component Selection

Here, we justify the selection of the two critical components of our framework, namely the inpainter $\mathcal{I}(\mathbf{x}, \mathbf{b}_i, c_j)$ and the ranker $\mathcal{R}(\hat{\mathbf{x}}_{ij}, c_j)$.

Experimental Setup. We use the OPA dataset [31] since it is the only one to contain human annotation for both positive and negative placements. We select

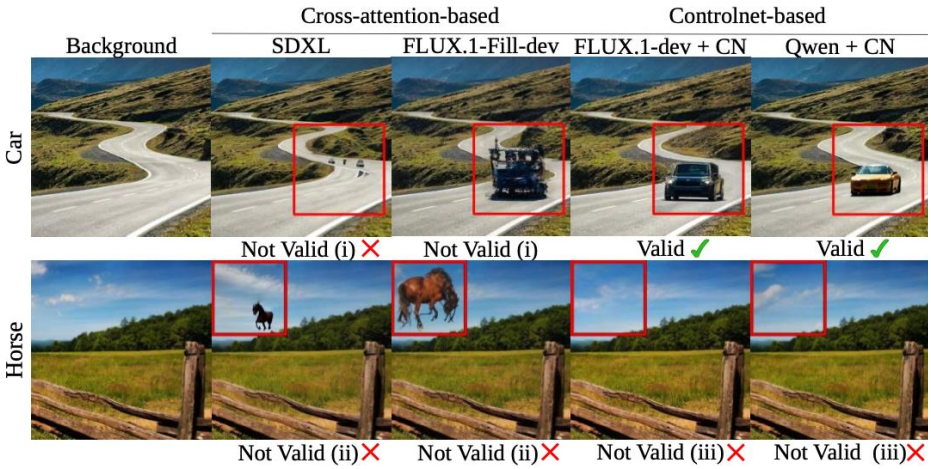


Fig. 6: Inpainter Comparison. We compare cross-attention-based inpainters with ControlNet-based variants. Cross-attention-based inpainters are prone to (i) inpainting artifacts or (ii) inpaint objects that don't respect scene semantics. While ControlNet-based methods more reliably enforce scene constraints: they either (iii) refuse implausible insertions or generate (valid) semantically coherent, context-aware objects.

464 entries (background, foreground object class). We run our full pipeline on these background images and collect our 1004 location annotations (both valid, and not valid). For each OPA annotation, we match it to the closest proposals of our pipeline, measured by IoU.

Ranker. As candidates for measuring human preference we compare different rankers: ImageReward [56], Aesthetic Score [46], CLIP Score [21], and VisionReward [55]. For a fair comparison, we fix the inpainter across all of them to FLUX [3]. In Fig. 5, we see that ImageReward and VisionReward better capture contextual realism, while CLIP prioritizes object class fidelity within the bounding box while neglecting global scene consistency, often failing to penalize implausible placements. In Tab. 2 we evaluate each ranker's ability to prioritize valid over invalid insertions using Average Precision (AP). ImageReward achieves the highest AP (57.1), outperforming CLIP and Aesthetics. This indicates that ImageReward aligns better to human judgments of plausible object placement.

Inpainter. We evaluate different inpainters of U-Net based diffusion (SDXL [38]) and DiT flow-matching backbones (FLUX [3], Qwen+CN). Fig. 6 shows that standard inpainters tend to fill the masked region with foreground content regardless of the surrounding scene semantics, e.g., SDXL [38] and FLUX [3] often place objects floating in the sky. Inversely, controlnet-based inpainters produce placements that are more consistent with scene depth and structure, matching perspective constraints and refusing impossible placements. In terms of ImageReward, we can see that Qwen+CN achieves the highest F1-score (53.5), for which we select it for our future experiments.

Table 2: Component Analysis. We measure inpainting and ImageReward performance **(Left)** We evaluate different rankers for the same inpainter FLUX. **(Right)** We evaluate different inpainters for the same Ranker. ImageReward and Qwen perform best. IR: ImageReward, CN: ControlNet.

Inpainter	Ranker	AP ↑
FLUX	Random	52.8
FLUX	Aesthetics	52.6
FLUX	CLIP Score	53.8
FLUX	**ImageReward**	**57.1**

Inpainter	Ranker	F1 ↑
SD	ImageReward	52.6
SD + CN	ImageReward	49.8
FLUX	ImageReward	48.5
FLUX + CN	ImageReward	40.1
Qwen + CN	ImageReward	**53.5**

Table 3: Downstream image editing quality evaluated by ImgEdit-Judge [57] (1-5). We compare our annotation pipeline against various placement strategies.

Method	Test set	PC ↑	VN ↑	PDC ↑	Avg ↑
Raw background	HiddenObjects	1.04	1.03	1.03	1.04
Full Mask	HiddenObjects	1.62	1.60	1.59	1.60
Random Bounding Box	HiddenObjects	2.73	2.62	2.62	2.65
Ours (Annotation Pipeline)	HiddenObjects	**3.83**	**3.63**	**3.63**	**3.69**
Raw background	OPA	1.00	1.00	1.00	1.00
Full Mask	OPA	1.66	1.66	1.66	1.66
Human Annotation	OPA	2.72	2.66	2.66	2.68
Random Bounding Box	OPA	2.97	2.84	2.84	2.89
Ours (Annotation Pipeline)	OPA	**4.05**	**3.83**	**3.83**	**3.90**

4.2 Image Editing Performance

Having established the optimal pipeline components, we now evaluate whether the resulting spatial priors translate into improved performance on the practical downstream task of object insertion via inpainting. A strong spatial prior should not only identify plausible locations but also actively guide inpainting models to produce photorealistic, human-aligned image composites.

Evaluation Setup. We evaluate object placement in both HiddenObjects and OPA. For fair comparison, we take a subset of backgrounds that have the same 29 shared COCO classes. We focus on the test set, and after filtering, we select 200 random images from HiddenObjects and all 76 OPA images. We use Qwen+CN for all our inpaintings. To test image editing performance we use the common benchmark of ImgEdit-Judge [57] which relies on Qwen2.5-VL [2] which evaluates placement quality giving a score between 1-5 by averaging the score of three different axes: Prompt Compliance (PC), Visual Naturalness (VN), and Physical and Detail Coherence (PDC).

Quantitative Analysis. In Tab. 3, we compare our inpainting performance on both test-sets against three baselines: (i) no edit, where the raw background is passed directly; (ii) full mask, covering the entire image; and (iii) random

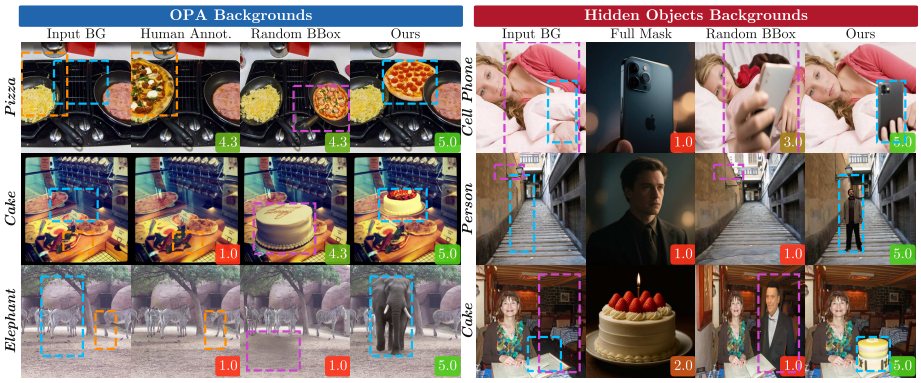


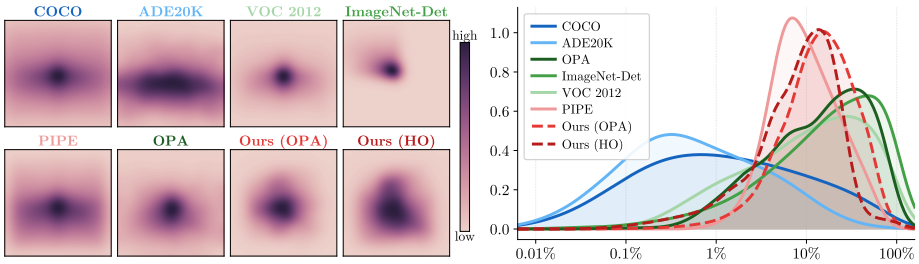
Fig. 7: Image inpainting with object placement priors. We evaluate four object placement approaches on the OPA [31] and HiddenObjects backgrounds. *Full Mask* inpaints the entire image. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random). Corner scores are ImgEdit-Judge [57]. Our pipeline consistently produces well-placed, photorealistic composites across object classes and image scenes, outperforming sparse human placement.

bounding box, uniformly sampled with area between 10% and 90% of the image and aspect ratio between 0.25 and 4. In OPA, we also compare with inpainting on the ground truth human annotation. Our method outperforms all baselines including OPA human annotation (3.90 vs. 2.68 on average). This clarifies that human annotation of object placement locations is inherently ambiguous, which is why human annotations yield similar performance to random proposals.

Qualitative Verification. Seeing the qualitative results in Fig. 7 we can verify the above observations. Inpainting over the full image (Full Mask) produces unrealistic composition. Random box is a bit better, by being contained to a specific region and scale, keeping the background realistic, yet producing no or unrealistic inpainting. More importantly, human annotations often fail to produce valid inpaintings, because such mask placements do not yield high quality diffusion inpainting, as happens with the insertion of a cake in the second line.

4.3 Systematic Biases Analysis

A fundamental challenge in object placement is that training datasets encode strong spatial and scale biases that models inevitably internalize. While our extracted spatial priors successfully capture human-aligned placement expectations, they are also likely to inherit systematic biases from the training dataset of both the used diffusion model and verifier. To this end, in this section we clarify how does our distribution compare to that of common object detection datasets such as COCO [29], VOC 2012 [14], ADE20K [64], ImageNet-Det [11] or other object placement datasets such as PIPE [52], OPA [31].



(a) Distribution of Object Centers **(b)** Density of Relative Object Area

Fig. 8: Analysis of Spatial and Scale Distributions. (a) Object center distributions across datasets: while traditional benchmarks like COCO [29], VOC [14], and ImageNet [11] exhibit a pronounced center-crop bias, our pipeline (evaluated on HiddenObjects and OPA backgrounds) yields a more diffuse distribution with higher spatial variance. **(b)** Distribution of bounding box areas: HiddenObjects and PIPE [52] lack small objects (0.01%–0.1% image area), which reveals a shared resolution bottleneck of underlying diffusion backbones in both generating or removing small objects.

Distribution of Object Centers. In Fig. 8a, we visualize the average distribution of center bounding box coordinates (similar to [18]). Standard datasets, such as COCO [29], VOC 2012 [14], PIPE [52], and ImageNet-Det [14], exhibit a pronounced center bias, concentrating object placements in a tight cluster around the image center. This is a well-known bias of how photographers naturally frame subjects. In contrast, our dataset displays the most spatially uniform spatial distribution, in comparison to both object placement datasets and object-centric recognition datasets. Still, unlike ADE20k, our spatial prior distribution closely reflects the aesthetic composition habits present in the large-scale web image datasets used to train diffusion models. For example, as seen in Fig. 4, while a bench could physically be placed anywhere on the ground, the aggregated prior exhibits a strong bottom-center bias.

Distribution of Object Scale. The object size distribution (Fig. 8b) reveals an equally important divergence. Older object-centric datasets (VOC [14] and Imagenet-Det [14]) are dominated by prominently large objects occupying 10–100% of the image area. In contrast, our pipeline concentrates on smaller relative sizes (1–10%), which is more consistent with the realistic placement of objects within complex scenes. Despite this shift toward realistic scales, the extracted priors still exhibit a bias toward medium-to-large bounding boxes. This stems from an intrinsic limitation of current diffusion models, which struggle to produce high-fidelity inpaintings when constrained to regions with very few pixels. Crucially, this resolution limit directly translates into a perspective bias. In high-depth scenes, due to perspective scene geometry, objects placed closer to the vanishing point must occupy a smaller pixel footprint. These small-scale generation attempts reliably yield low-quality completions that fail to pass the verifier's detection thresholds, resulting in a sparsity of background placements.

Table 4: Comparison of object placement baselines on the HiddenObjects test set. Our baseline distilled model outperforms zero-shot VLMs and other methods for object placement trained on OPA.

Method	Train Set	mAP (%)	IoU50@1 (%)	IoU@1 (%)	IoU50@5 (%)	IoU@5 (%)
Qwen2.5-VL-72B	Zero-shot	1.3	7.9	14.0	12.7	18.8
Qwen2.5-VL-7B	Zero-shot	1.4	8.6	13.9	10.1	16.5
InternVL3-8B	Zero-shot	0.3	3.3	11.2	9.2	18.7
LLaVA-OneVision-7B	Zero-shot	1.0	11.7	20.9	16.3	24.7
LLaVA-1.5-7B	Zero-shot	1.0	6.4	17.0	8.8	22.8
LLaVA-1.5-13B	Zero-shot	1.2	6.4	14.0	7.5	15.6
BootPlace [66]	Cityscapes	0.1	0.9	6.0	0.9	6.0
TerseNet [50]	OPA	1.4	20.9	29.5	20.9	29.5
GracoNet [67]	OPA	2.1	23.7	35.9	23.7	35.9
PlaceNet [60]	OPA	2.9	43.5	42.4	43.5	42.4
Ours	OPA	28.0	36.4	33.0	51.9	57.6
Ours	HiddenObjects	**56.6**	**62.9**	**55.2**	**79.1**	**67.7**

5 Model Distillation for Object Placement

To quickly infer placement proposals on unseen images for downstream applications, we distill our extracted spatial priors into a lightweight model.

Architecture. We formulate object placement as a multi-class, closed-set prediction task. We extract multi-scale visual features of background scenes using a frozen ResNet-50 backbone. These features are processed by a DETR-style Transformer encoder-decoder. The encoder consists of 6 layers, while the decoder utilizes 50 object-conditioned learnable queries per image, formed by adding query offsets to the embedded target class. For the final predictions, we use a 3-layer MLP bounding box head that regresses normalized coordinates, and a 2-layer MLP plausibility head that regresses the ImageReward. We train using Hungarian matching with a regression loss (L1 + GIoU) explicitly weighted by the normalized reward scores. To ensure high-quality supervision, we train using the top 20 bounding boxes of our spatial prior with the highest reward score. We optimize using AdamW with a learning rate of 1e−4, a weight decay of 1e−4, and clip gradients with a norm higher than 0.1. We train at a batch size of 128 for maximum 200 epochs using early stopping with a patience of 15 epochs.

Experimental Setup. We evaluate our model in both OPA and HiddenObjects. For fair cross-dataset evaluation, we use the subset of HiddenObjects that contains the 28 shared foreground object classes across existing placement. This results in 9,438 train and 504 test (565,948 location annotations). We also train on the subset of OPA that has the same 28 shared classes of 958 training images. We measure performance of bounding box proposals using Intersection over Union (IoU) and Mean Average Precision (mAP). We also measure IoU50

Table 5: We compare object insertions on OPA using ImgEdit-Judge. A model trained on HiddenObjects (ours) outperforms both the original OPA human annotations as well as a model trained on OPA. The slow pipeline of Sec. 3, provides the best results.

	Method	PC. ↑	VN. ↑	PDC. ↑	Avg. ↑
	OPA Human Annotation	2.72	2.66	2.66	2.68
Ours	Model trained on OPA	3.39	3.33	3.33	3.35
Ours	Model trained on HiddenObjects	3.53	3.38	3.38	3.43
Ours	**Diffusion Annotation (Sec. 3)**	**4.05**	**3.83**	**3.83**	**3.90**

accuracy where predictions achieving a maximum IoU ≥ 0.50 against ground-truth are considered positive hit. For more details, see Appendix C.

Evaluation. We benchmark our distilled model against state-of-the-art generalist VLMs (including Qwen2.5-VL-72B [2], InternVL3-8B [9], and LLaVA-1.5 [30]) and the dedicated off-the-shelf object placement models PlaceNet [60], GracoNet [67], TerseNet [50] and BootPlace [66]. In Tab. 4, we observe that zero-shot VLMs fail (peak around 1.4% mAP), and existing placement models struggle beyond top-1 localization as they have only been trained on sparse spatial priors. In contrast, our model performs best with 56.6% mAP and 62.9% IoU50@1. Similar to Sec. 4.2, we also evaluate the model's performance in object placement (Tab. 5). Our model trained on HiddenObjects performs better than the Human Annotations of OPA, as well as a model trained on OPA, proving that our distillation process can lead to high quality bounding box proposals.

Inference Time. To justify the need for distillation, we report measurements on an H100 GPU. The distilled model achieves a latency of 3.77 ms/image on average tested on 100 images with 188.9 MB peak GPU memory. Compared to 14.49 minutes/image and 65.9 GB for the full pipeline, this yields a 230,000× speedup and 300× memory reduction.

6 Conclusion

We presented a scalable framework for extracting class-conditioned spatial priors from diffusion models, bypassing the limitations of sparse manual annotations and artifact-prone synthetic datasets. Our pipeline produces HiddenObjects, a dataset of 27M evaluated placements across 27k backgrounds that encodes human-aligned placement statistics. Distilling these priors into a lightweight model enables real-time inference, reducing computation by over five orders of magnitude compared to the full pipeline. We outperform both dedicated placement models and zero-shot VLMs, and, notably, surpass sparse human annotations in downstream image editing quality. Our analysis further reveals that, while the extracted priors inherit certain photographic composition biases from their generative supervisors, they successfully enforce complex structural and scene constraints. We believe these explicit spatial priors offer a useful foundation for controllable image editing, data augmentation, and scene understanding.

7 Acknowledgments

Dim P. Papadopoulos was supported by the DFF Sapere Aude Starting Grant "ACHILLES". This work was partly supported by the Pioneer Centre for AI, DNRF grant number P1. We thank Mehmet Onurcan Kaya, Thanos Delatolas, and Nico Lang for insightful feedback and discussions.

References

1. Abdelreheem, A., Aleotti, F., Watson, J., Qureshi, Z., Eldesokey, A., Wonka, P., Brostow, G., Vicente, S., Garcia-Hernando, G.: Placeit3d: Language-guided object placement in real 3d scenes. In: ICCV (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), https://arxiv.org/abs/2502.13923
3. Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arxiv **abs/2506.15742** (2025)
4. Bau, D., Zhu, J.Y., Strobelt, H., Lapedriza, A., Zhou, B., Torralba, A.: Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences (2020). https://doi.org/10.1073/pnas.1907375117, https://www.pnas.org/content/early/2020/08/31/1907375117
5. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: Gan dissection: Visualizing and understanding generative adversarial networks. In: ICLR (2019)
6. Boukerche, A., Zheng, L., Alfandi, O.: Outlier detection: Methods, models, and classification. ACM Comput. Surv. **53**(3), 55:1–55:37 (2021)
7. Canet Tarrés, G., Lin, Z., Zhang, Z., Zhang, J., Song, Y., Ruta, D., Gilbert, A., Collomosse, J., Kim, S.Y.: Thinking outside the bbox: Unconstrained generative object compositing. In: ECCV. pp. 476–495 (2024)
8. Chen, F., Shen, Y., Xu, L., Yuan, Y., Zhang, S., Niu, Y., Wen, L.: Referring layer decomposition. arXiv preprint arXiv:2602.19358 (2026)
9. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: CVPR. pp. 24185–24198 (2024)
10. Cheng, X., Zhai, P., Yang, D., Meng, X., Xia, Y., Zhang, L.: Diverse object placement with dual interaction. Neurocomputing p. 131161 (2025)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
12. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV. pp. 1422–1430 (2015)
13. Dvornik, N., Mairal, J., Schmid, C.: Modeling visual context is key to augmenting object detection datasets. In: ECCV. pp. 364–380 (2018)
14. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

15. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR (2008)
16. Gao, B., Zhang, B., Niu, L.: Object placement for anything. ICME (2025)
17. Gao, J., Joseph, K.J., la Torre, F.D.: Teleportraits: Training-free people insertion into any scene. In: ICCV (2025)
18. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
19. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Transactions on graphics (TOG) **26**(3), 4–es (2007)
20. He, J., Li, W., Liu, Y., Kim, J., Wei, D., Pfister, H.: Affordance-aware object insertion via mask-aware dual diffusion. arXiv preprint arXiv:2412.14462 (2024)
21. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) EMNLP. pp. 7514–7528. Association for Computational Linguistics (2021)
22. Huang, I., Bao, Y., Truong, K., Zhou, H., Schmid, C., Guibas, L., Fathi, A.: Fireplace: Geometric refinements of llm common sense reasoning for 3d object placement. In: CVPR (2025)
23. Kulal, S., Brooks, T., Aiken, A., Wu, J., Yang, J., Lu, J., Efros, A.A., Singh, K.K.: Putting people in their place: Affordance-aware human insertion into scenes. In: CVPR. pp. 17089–17099 (2023)
24. Lee, D., Liu, S., Gu, J., Liu, M.Y., Yang, M.H., Kautz, J.: Context-aware synthesis and placement of object instances. NeurIPS **31** (2018)
25. Li, C., Wang, W., Li, Q., Lepri, B., Sebe, N., Nie, W.: Freeinsert: Disentangled text-guided object insertion in 3d gaussian scene without spatial priors. ACM Multimedia (2025)
26. Li, T., Ku, M., Wei, C., Chen, W.: Dreamedit: Subject-driven image editing. Trans. Mach. Learn. Res. (2023)
27. Liang, D., Jia, J., Liu, Y., Lau, R.W.: Hocomp: Interaction-aware human-object composition. arXiv preprint arXiv:2507.16813 (2025)
28. Lin, C.H., Yumer, E., Wang, O., Shechtman, E., Lucey, S.: St-gan: Spatial transformer generative adversarial networks for image compositing. In: CVPR. pp. 9455–9464 (2018)
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)
30. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR. pp. 26296–26306 (2024)
31. Liu, L., Liu, Z., Zhang, B., Li, J., Niu, L., Liu, Q., Zhang, L.: Opa: object placement assessment dataset. arXiv preprint arXiv:2107.01889 (2021)
32. Lu, L., Li, J., Zhang, B., Niu, L.: Dreamcom: Finetuning text-guided inpainting model for image composition. arXiv preprint arXiv:2309.15508 (2023)
33. Niu, L., Liu, Q., Liu, Z., Li, J.: Fast object placement assessment. arXiv preprint arXiv:2205.14280 (2022)
34. Papadopoulos, D.P., Tamaazousti, Y., Ofli, F., Weber, I., Torralba, A.: How to make a pizza: Learning a compositional layer-based gan model. In: CVPR (2019)
35. Parihar, R., Gupta, H., VS, S., Babu, R.V.: Text2place: Affordance-aware text guided human placement. In: ECCV. pp. 57–77 (2024)
36. Parihar, R., Sarkar, S., Vora, S., Kundu, J., Babu, R.V.: Monoplace3d: Learning 3d-aware object placement for 3d monocular detection. In: CVPR (2025)

37. Petersen, J., Abati, D., Habibian, A., Wiggers, A.: Scene-aware location modeling for data augmentation in automotive object detection. arXiv preprint arXiv:2504.17076 (2025)
38. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. In: ICLR (2024)
39. Poska, M., Huang, S.X., Hwang, B.: Hopnet: Harmonizing object placement network for realistic image generation via object composition. In: CVPR. pp. 6344–6354 (2025)
40. Qin, Y., Xu, J., Wang, R., Chen, X.: Think before placement: Common sense enhanced transformer for object placement. In: ECCV. pp. 35–50 (2024)
41. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
42. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
43. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024)
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2021)
45. Schouten, M., Kaya, M.O., Belongie, S., Papadopoulos, D.P.: Poem: Precise object-level editing via mllm control. In: SCIA (2025)
46. Schuhmann, C.: Aesthetic predictor v2.5. `https://github.com/discus0434/aesthetic-predictor-v2-5/` (may 2024), gitHub repository
47. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)
48. Tewel, Y., Gal, R., Samuel, D., Atzmon, Y., Wolf, L., Chechik, G.: Add-it: Training-free object insertion in images with pretrained diffusion models. In: ICLR (2025), `https://openreview.net/forum?id=ZeaTvXw080`
49. Torralba: Context-based vision system for place and object recognition. In: ICCV. pp. 273–280 (2003)
50. Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J.M., Chari, V.: Learning to generate synthetic data via compositing. In: CVPR. pp. 461–470 (2019)
51. Volokitin, A., Susmelj, I., Agustsson, E., Van Gool, L., Timofte, R.: Efficiently detecting plausible locations for object placement using masked convolutions. In: ECCV Workshop. pp. 252–266 (2020)
52. Wasserman, N., Rotstein, N., Ganz, R., Kimmel, R.: Paint by inpaint: Learning to add image objects by removing them first. In: CVPR (2025)
53. Winter, D., Shul, A., Cohen, M., Berman, D., Pritch, Y., Rav-Acha, A., Hoshen, Y.: Objectmate: A recurrence prior for object insertion and subject-driven generation. In: ICCV. pp. 16281–16291 (2025)
54. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
55. Xu, J., Huang, Y., Cheng, J., Yang, Y., Xu, J., Wang, Y., Duan, W., Yang, S., Jin, Q., Li, S., Teng, J., Yang, Z., Zheng, W., Liu, X., Zhang, D., Ding, M., Zhang, X., Huang, S., Gu, X., Huang, M., Tang, J., Dong, Y.: Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In: AAAI. pp. 11269–11277 (2026)

56. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: learning and evaluating human preferences for text-to-image generation. In: NeurIPS. pp. 15903–15935 (2023)
57. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. arXiv preprint arXiv:2505.20275 (2025)
58. Yuan, L., Hong, J., Sarukkai, V., Fatahalian, K.: Learning to place objects into scenes by hallucinating scenes around objects. In: NeurIPS (2023)
59. Yun, J., Abati, D., Omran, M., Choo, J., Habibian, A., Wiggers, A.: Imagining the unseen: Generative location modeling for object placement (2025), `https://arxiv.org/abs/2410.13564`
60. Zhang, L., Wen, T., Min, J., Wang, J., Han, D., Shi, J.: Learning object placement by inpainting for compositional data augmentation. In: ECCV. pp. 566–581 (2020)
61. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: CVPR. pp. 3836–3847 (2023)
62. Zhang, S.H., Zhou, Z.P., Liu, B., Dong, X., Hall, P.: What and where: A context-based recommendation system for object insertion. Computational Visual Media **6**(1), 79–93 (2020)
63. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)
64. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV **127**(3), 302–321 (2019)
65. Zhou, H., Ma, R., Zhang, L.X., Gao, L., Mahdavi-Amiri, A., Zhang, H.: Sac-gan: Structure-aware image composition. IEEE Transactions on Visualization and Computer Graphics **30**(7), 3151–3165 (2022)
66. Zhou, H., Zuo, X., Ma, R., Cheng, L.: Bootplace: Bootstrapped object placement with detection transformers. In: CVPR. pp. 19294–19303 (2025)
67. Zhou, S., Liu, L., Niu, L., Zhang, L.: Learning object placement via dual-path graph completion. In: ECCV. pp. 373–389 (2022)
68. Zhu, S., Lin, Z., Cohen, S., Kuen, J., Zhang, Z., Chen, C.: Topnet: Transformer-based object placement network for image compositing. In: CVPR. pp. 1838–1847 (2023)

Appendix

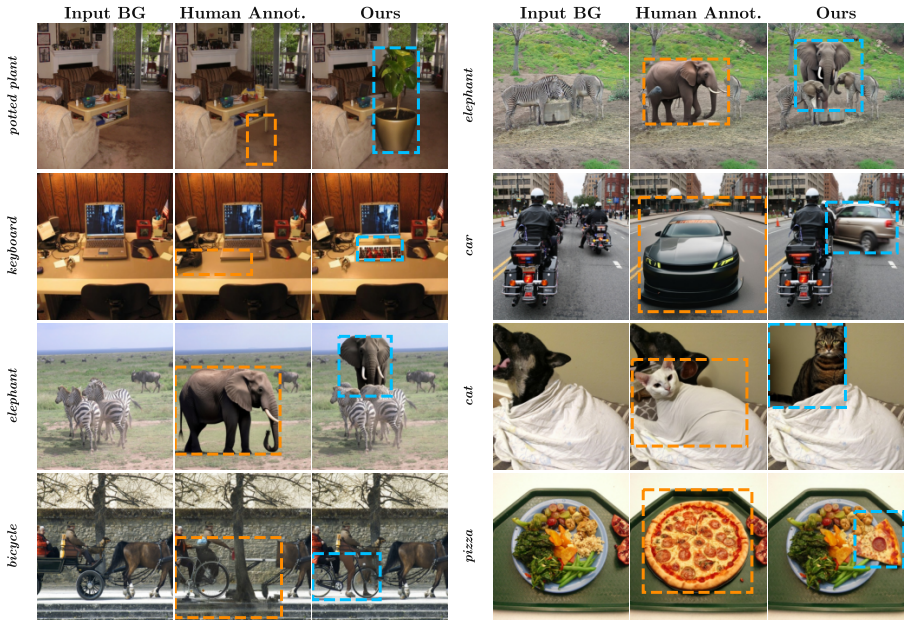


Fig. 9: Image inpainting with object placement priors. We evaluate three object placement approaches on the OPA backgrounds. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random).

A Image Editing

In this section, we provide extended qualitative results for the image editing and object placement experiments originally presented in **Fig. 7**. Specifically, we present additional comparisons of object insertions guided by bounding box proposals from three distinct sources: Human annotations, Randomly sampled bounding boxes, and the extracted spatial priors (Ours). For our method, we select the top-1 annotation ranked by ImageReward. To avoid replacing dominant objects already present in the background, the selected bounding box must have an Intersection over Union (IoU) of less than 50% with the largest object detected in the background scene by Grounded-DINO, an open-set object detector. We illustrate a curated set of these qualitative inpainting results in Fig. 9, and provide an extensive gallery of further comparisons in Figs. 17–20.

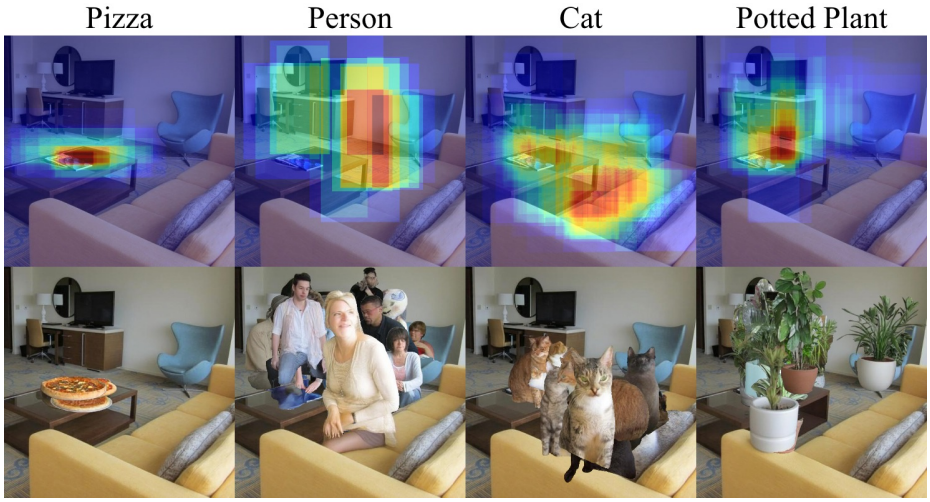


Fig. 10: Composite Overlay of Inpaintings. Multiple valid inpaintings are aggregated into a single composite scene. Accepted objects are segmented and overlaid on the source background. Insertions are ordered by preference ranking so the highest-ranked object remains unoccluded at the top of the stack.

B Spatial Prior Visualization

Building upon the aggregated spatial prior distributions introduced in **Fig. 4** of the main paper, we present a comprehensive set of randomly sampled spatial prior visualizations across 25 distinct object categories. These qualitative examples are detailed in Figs. 14–16. To construct the heatmaps, ranking scores are first normalized across candidate boxes using a softmax, after which the resulting weights are accumulated over the pixels contained within each bounding box to form a spatial density map. The aggregated map is then min–max normalized and blended with the input image to produce the final visualization overlay. We aggregate only those bounding box annotations that achieve a verification detector confidence greater than 0.4 and a positive preference ranker score ($r_{ij} > 0$). This robust filtering step ensures that the resulting spatial priors strictly reflect valid, highly preferred, and visually coherent object placements. In Fig. 10, we complement the heatmaps of **Fig. 1** with an overlay composite of the segmented objects of multiple inpaintings, overlayed in a lower-to-higher ranking order.

C Placement Model

As discussed in the main paper, the distillation model we use to learning object placement from our data is a simple variant of DETR. Here, we provide detailed information about our architecture and loss function.

Architecture. Our architecture is visualized in Fig. 11. The input is an RGB image tensor of center-cropped images (512×512). It is fed to a DETR model

that consists of a transformer encoder-decoder with ResNet-50 image feature patches and sinusoidal positional encoding. The transformer encoder processes the sequence using 6 layers with hidden dimension 256, 8 attention heads, and feedforward layers of size 2048. An input category (e.g., "apple", "cat") is then mapped to a set of learnable class embeddings to an embedding vector $c \in \mathbb{R}^{B\times 256}$. This vector is replicated across $|Q| = 50$ queries, which correspond to placement proposals, and is added to learnable queries to produce the final $\mathbb{R}^{B\times 50\times 256}$ input to DETR.

We apply two prediction heads to the transformer outputs, which correspond to $|Q|$ decoded queries. The first is a bounding box head implemented as an MLP consisting of three connected layers, with hidden dimension of 256 and ReLU activation functions, mapping to 4 output values being the top left coordinate of the bounding box in normalized image coordinates followed by its width and height. In parallel, another MLP with the same architecture regresses one parameter that corresponds to the reward score. During inference, the model produces 50 candidate placements per image. The predictions are ranked using the sigmoid of the plausibility score.

Loss. We first perform bipartite matching between predictions and ground-truth boxes using the Hungarian algorithm, minimizing a cost function $\mathcal{C}$ that combines L_1 distance and Generalized Intersection over Union (GIoU). For matched pairs, the spatial loss is defined as:

$$\mathcal{L}_{\text{bbox}} = \omega \left[\lambda_{L1}\|b - \hat{b}\|_1 + \lambda_{giou}(1 - \text{GIoU}(b, \hat{b}))\right] \tag{6}$$

where $\omega = 0.5 + 0.5R$ is a weighting factor derived from the image-reward score R. The reward head is trained to regress the maximum IoU of each query relative to the ground-truth set. Let $\hat{p}_j$ be the logit for query j. The plausibility loss is:

$$\mathcal{L}_{\text{plaus}} = \frac{1}{N}\sum_{j=1}^{N} \|\sigma(\hat{p}_j) - \max_k \text{IoU}(\hat{b}_j, b_k)\|^2 \tag{7}$$

Our final objective is $\mathcal{L} = \mathcal{L}_{\text{bbox}} + 0.5\mathcal{L}_{\text{plaus}}$, with $\lambda_{L1} = 5.0$ and $\lambda_{giou} = 2.0$.

D Inpainting Speedup

The main computational bottleneck in computing spatial priors is the diffusion-based inpainting stage. For each image, we evaluate 1,004 candidate boxes, only a small number of which correspond to valid insertions (i.e., confidence greater than 0.4). As discussed in the main paper, to detect that, we introduce an early-stopping detection process based on divergence in denoising performance in the early denoising steps. Below, we detail our early stopping heuristic derivation.

Criterion. Given a diffusion model ε_θ that denoises a noisy image $\mathbf{x}_t$ at step t by computing the noise $\varepsilon_\theta(\mathbf{x}_t, c)$ for condition c. An image is inpainted by taking the difference between the conditional branch, e.g., $c =$ "cat" and the

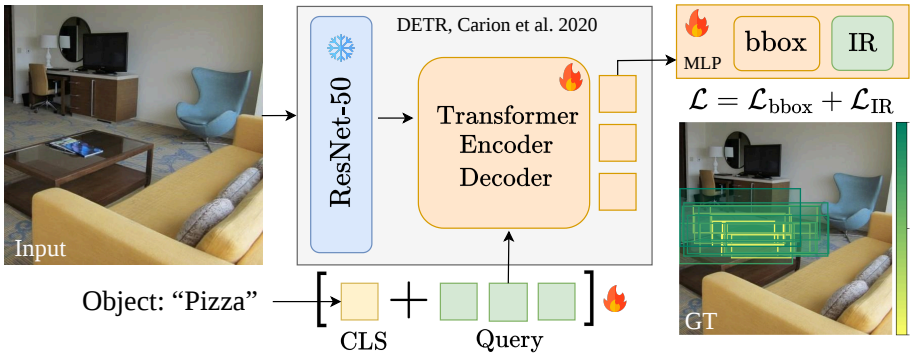


Fig. 11: Placement Distillation Model. Given a background image and a target object class, the model predicts plausible insertion locations using a DETR-style architecture. The transformer decoder operates on a fixed set of learned queries that are conditioned on the target class embedding. Each query predicts a candidate bounding box with its plausibility score. During inference, predicted boxes are ranked by their plausibility scores to obtain the most likely placement locations for the target object.

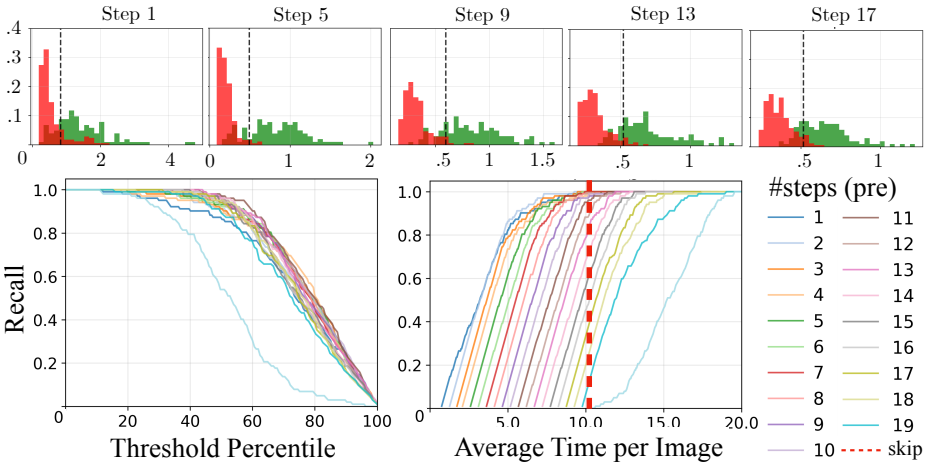


Fig. 12: Speedup Analysis. Top: Distribution of $\Delta(\mathbf{x}_t, \varnothing)$ for 10040 inpaintings. **Bottom:** Plotting recall versus time across time steps shows that using only the second denoising iteration seems to be sufficient to get a 2x speed-up improvement on an 80% recall. The dotted red line denotes the cost of skipping the first step all along $N \cdot \tau_{20}$.

unconditional branch $c = \varnothing$. Our observation is simple: we measure how much these two noise estimates diverge throughout training:

$$\Delta(\mathbf{x}_t, \varnothing) = \|\varepsilon_\theta(\mathbf{x}_t, c) - \varepsilon_\theta(\mathbf{x}_t, \varnothing)\|_1. \tag{8}$$

In the top part of Fig. 12, we compute this distribution per denoising step for 15 random images and observe how the distribution of $\Delta(\mathbf{x}_t, \varnothing)$ changes

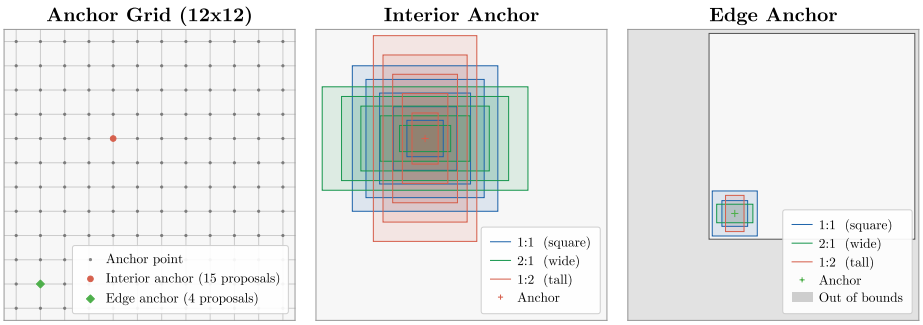


Fig. 13: Bounding Box Proposals. Left: The 12×12 anchor grid. **Middle**: Full coverage set for an interior anchor. **Right**: Pruned set for a corner anchor due to boundary constraints.

across timestep. We notice that there is a fine separation between successful and unsuccessful generations, where successful ones exhibit much higher values (mean 1.47) than failed ones (mean 0.38). While in step-5 generations are better separated to speedup computation we have to trade separability with efficiency.

Threshold computation. If N is the number of boxes and $s(r_\lambda)$ is the percentage of successful boxes for a given recall r calculated for a threshold λ and τ_t is the computational time required until step t, the total computational time per recall $\tau(r_\lambda)$:

$$\tau(r_\lambda) = N(\tau_t + s(r_\lambda)\tau_{20}), \tag{9}$$

where the r.h.s. corresponds to the check, while the second to the full inpainting. As shown in Fig. 12, we find that a calibrated threshold $\lambda = 0.7148$ achieves 81% recall for two steps, with only $s(r_\lambda) = 0.3$ fraction of bounding box locations being passed to the next iteration, leading to a 2.4x reduction.

E Bounding box Proposals

Bounding box proposals for inpainting are produced using a sliding window approach, explained in Fig. 13. Given an image of size $S \times S$ and a target number of proposals N, we first derive a square grid of $G \times G$ anchor points, where $G = \lfloor \sqrt{\lceil N/3 \rceil} \rfloor$. These anchors are uniformly spaced with a half-stride offset to provide symmetric padding from the image boundaries.

For each anchor, we generate candidate boxes by combining three aspect ratios $\mathcal{R} = \{1:1, 2:1, 1:2\}$ with five linearly spaced scales $s \in [s_{\min}, s_{\max}]$. To ensure area-equivalence across ratios, the width and height are computed as $w = s\sqrt{r_w/r_h}$ and $h = s\sqrt{r_h/r_w}$. A candidate box $\mathbf{b}$ is added to the final proposal set $\mathcal{B}$ only if it is strictly contained within the image manifold $[0, S] \times [0, S]$.

We set $S = 512$, $N = 435$, and a scale range of $[64, 256]$. While the unconstrained maximum number of boxes is 2,160 (144 anchors $\times$ 15), the boundary constraint prunes edge-adjacent candidates to consider only fully visible objects. This deterministic configuration yields a final set of $|\mathcal{B}| = 1,004$ bounding boxes.

F Background - Foreground Assignment

To produce general enough coverage of object placement, we define 10 macro-categories of objects: person, vehicles, outdoor items, animals, sport items, kitchen items, food, furniture, technology, and miscellaneous. Then, we manually aggregate 50 COCO classes per macro-category. Finally, we employ a manually defined object-to-scene taxonomy between the 50 foreground object classes from COCO and the 126 distinct background classes from Places365 to minimize non-meaningful object-background placement requests. The full assignment is detailed in Tab. 6.

Table 6: Assignment between COCO foreground and Places365 background classes.

Category	COCO Foreground	Places365 Background
person	person	alley, amphitheater, apartment building, atrium, attic, ballroom, desert sand, downtown, market, martial arts gym, plaza, shopping mall, street, temple, wheat field
person	umbrella	attic, balcony exterior, balcony interior, boardwalk, campus, dining hall, patio, street
person	suitcase	airport terminal, atrium, attic, bus station, car interior, hotel room, motel
person	handbag	atrium, attic, car interior, coffee shop, department store, restaurant, shopping mall
person	backpack	airport terminal, atrium, attic, campus, car interior, forest path, park
vehicles	bicycle	attic, campus, downtown, park, promenade, residential neighborhood
vehicles	car	auto factory, desert road, garage, gas station, highway, parking lot, street
vehicles	motorcycle	desert road, garage, gas station, highway, parking lot, street
vehicles	airplane	airfield, hangar indoor, hangar outdoor, runway, sky
vehicles	boat	boat deck, boathouse, harbor, lagoon, lake, ocean, river
outdoor	traffic light	downtown, driveway, promenade, runway, street
outdoor	bench	courtyard, park, patio, promenade, residential neighborhood
outdoor	fire hydrant	downtown, park, residential neighborhood, runway, street
outdoor	stop sign	courtyard, crosswalk, downtown, residential neighborhood, street
outdoor	parking meter	downtown, parking lot, promenade, residential neighborhood, street
animals	dog	alley, botanical garden, chalet, field wild, kennel, park, residential neighborhood, veterinarians office, yard
animals	cat	alley, balcony, bedroom, botanical garden, chalet, house, living room, veterinarians office, yard
animals	elephant	bamboo forest, desert sand, field wild, forest path, rainforest
animals	horse	cliff, farm, field wild, kennel, pasture, stable, tundra
animals	cow	barn, corn field, farm, field wild, kennel, pasture
sports	kite	athletic field, beach, chalet, field wild, mountain path, park
sports	skateboard	campus, chalet, garage, park, playground, promenade, street
sports	tennis racket	apartment building, athletic field, chalet, gymnasium, park, playground
sports	surfboard	beach, boat deck, coast, garage, pier, wave
sports	sports ball	athletic field, baseball field, basketball court, gymnasium, park, soccer field
kitchen	bottle	bar, cafeteria, coffee shop, dining room, kitchen, restaurant, restaurant patio
kitchen	cup	cafeteria, coffee shop, dining room, fastfood restaurant, kitchen, restaurant
kitchen	wine glass	banquet hall, bar, coffee shop, dining room, restaurant, restaurant patio
kitchen	spoon	bar, cafeteria, coffee shop, dining room, kitchen, restaurant
kitchen	fork	bar, cafeteria, coffee shop, dining room, kitchen, restaurant
food	sandwich	cafeteria, coffee shop, fastfood restaurant, picnic area, restaurant, restaurant patio
food	apple	dining room, kitchen, market, picnic area, vegetable garden
food	cake	bakery shop, banquet hall, cafeteria, dining room, picnic area, restaurant
food	pizza	dining room, fastfood restaurant, picnic area, pizzeria, restaurant
food	orange	dining room, kitchen, market, picnic area, vegetable garden
furniture	potted plant	attic, balcony interior, formal garden, living room, office, patio, zen garden
furniture	chair	attic, basement, beach house, bedroom, classroom, dining room, living room, office
furniture	couch	attic, basement, beach house, hotel room, house, living room, reception, television room
furniture	dining table	banquet hall, dining room, kitchen, restaurant, restaurant patio
furniture	bed	attic, bedchamber, bedroom, dorm room, hotel room
technology	tv	attic, basement, bedroom, home theater, hotel room, living room, television room
technology	laptop	attic, basement, bedroom, dorm room, home office, library, office
technology	cell phone	basement, bedroom, dorm room, home office, office, reception
technology	keyboard	basement, bedroom, computer room, dorm room, home office, office
technology	microwave	basement, cafeteria, dining room, kitchen, office, restaurant kitchen
misc.	book	attic, basement, bookstore, classroom, dorm room, library, living room, office
misc.	toothbrush	bathroom
misc.	clock	attic, basement, berth, classroom, house, living room, office, reception
misc.	scissors	art studio, artists loft, basement, classroom, fabric store, home office, office
misc.	vase	archive, attic, basement, bedroom, house, living room, reception

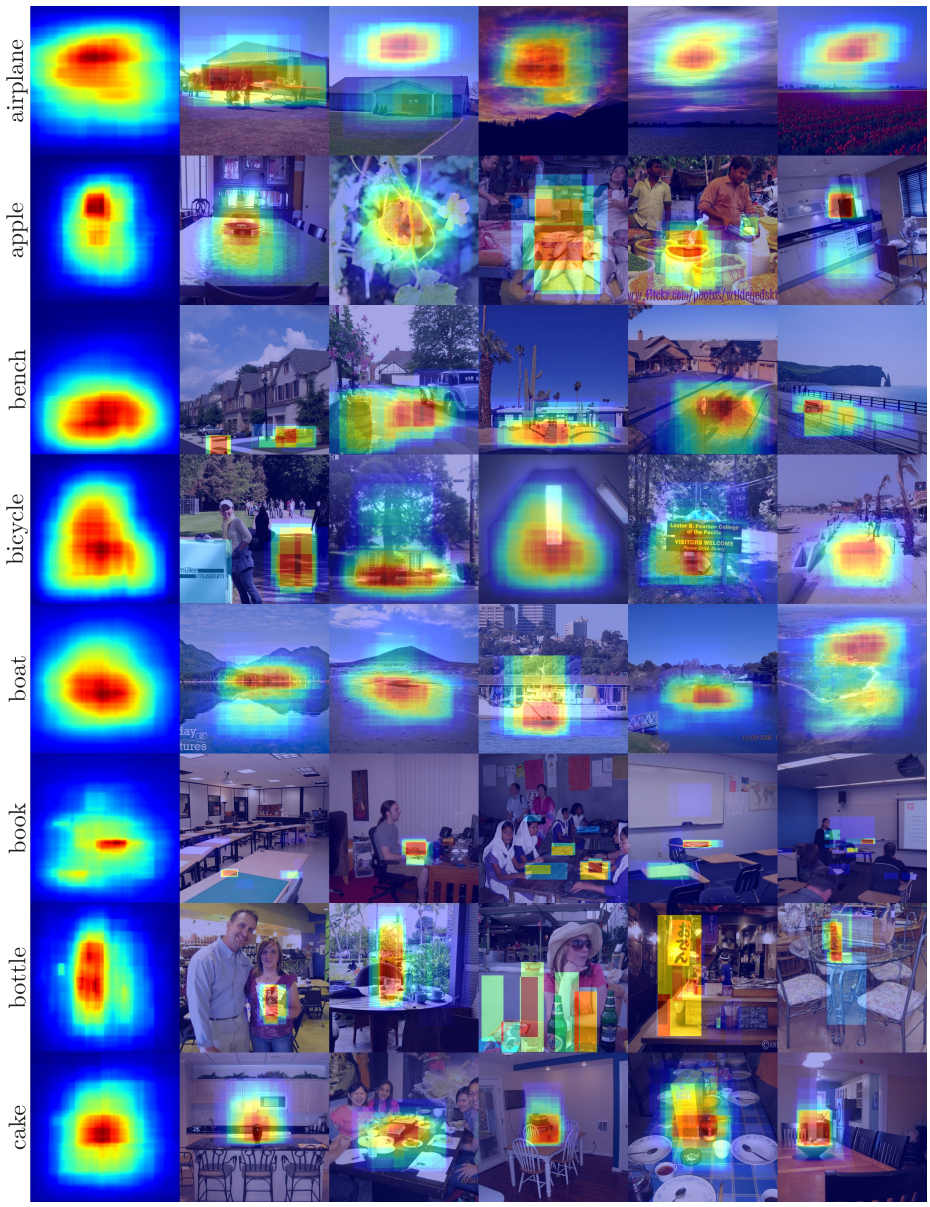


Fig. 14: Spatial Prior. Object placements are computed for the following classes: airplane, apple, bench, bicycle, boat, book, bottle, cake.

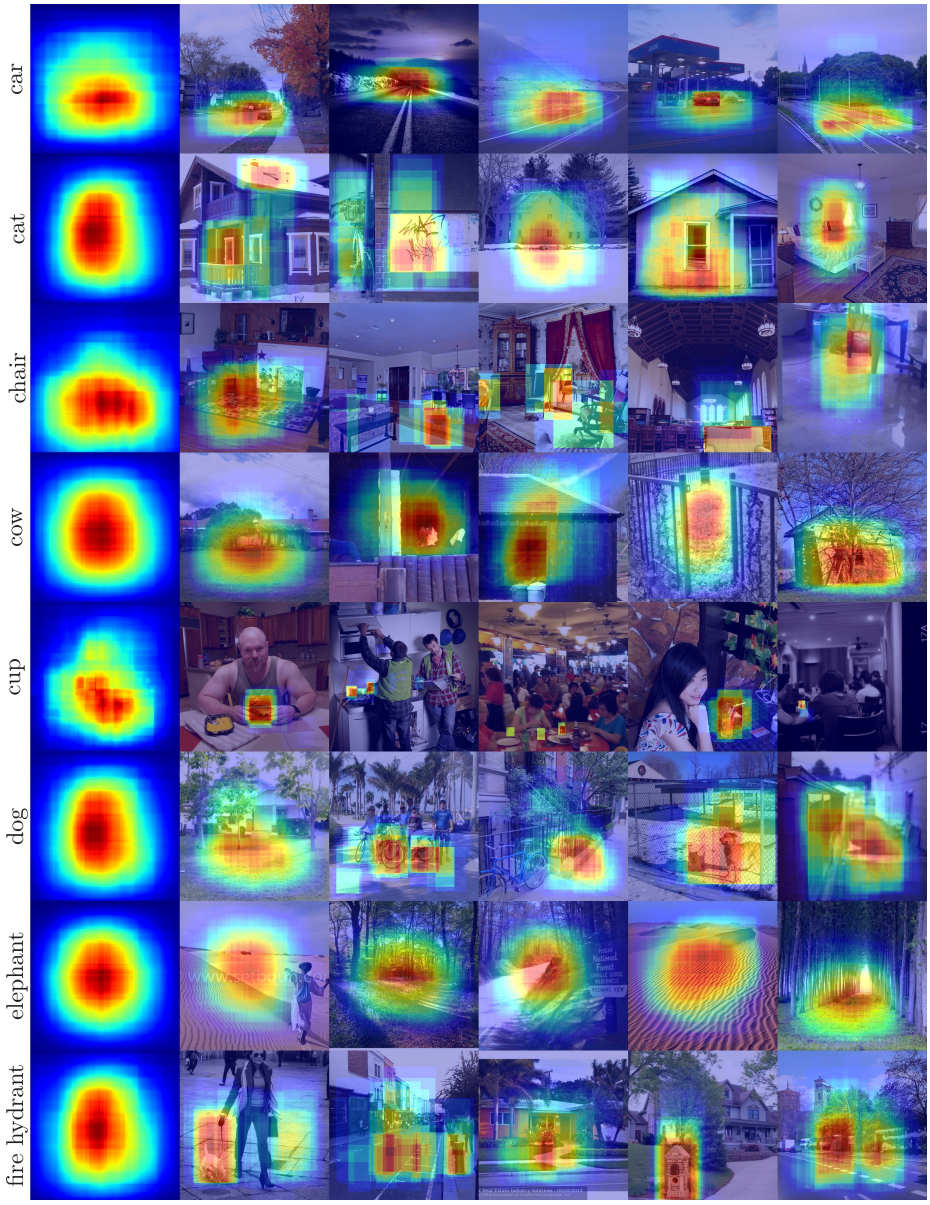


Fig. 15: Spatial Prior. Object placements are computed for the following classes: car, cat, chair, cow, cup, dog, elephant, fire hydrant.

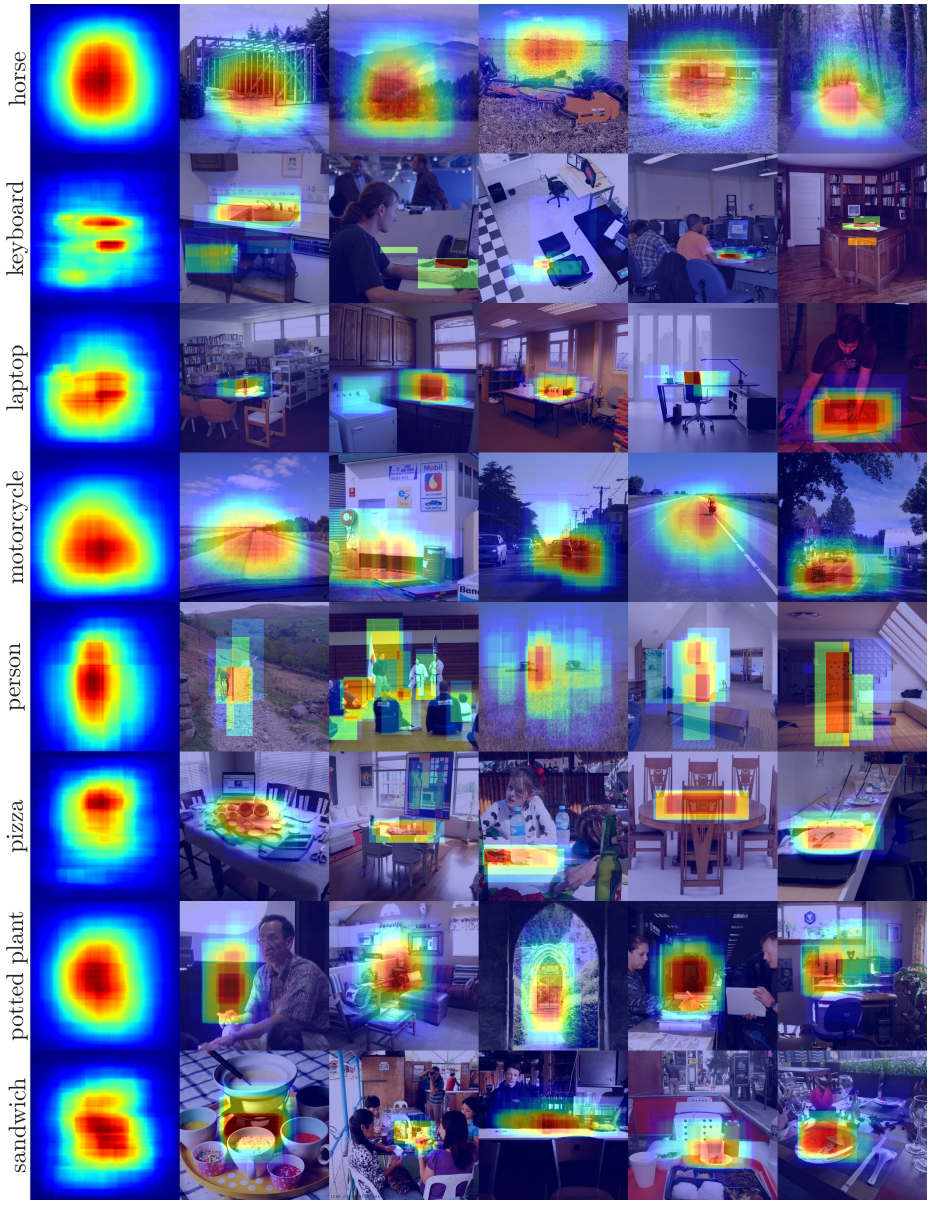


Fig. 16: Spatial Prior. Object placements are computed for the following classes: horse, keyboard, laptop, motorcycle, person, pizza, potted plant, sandwich.

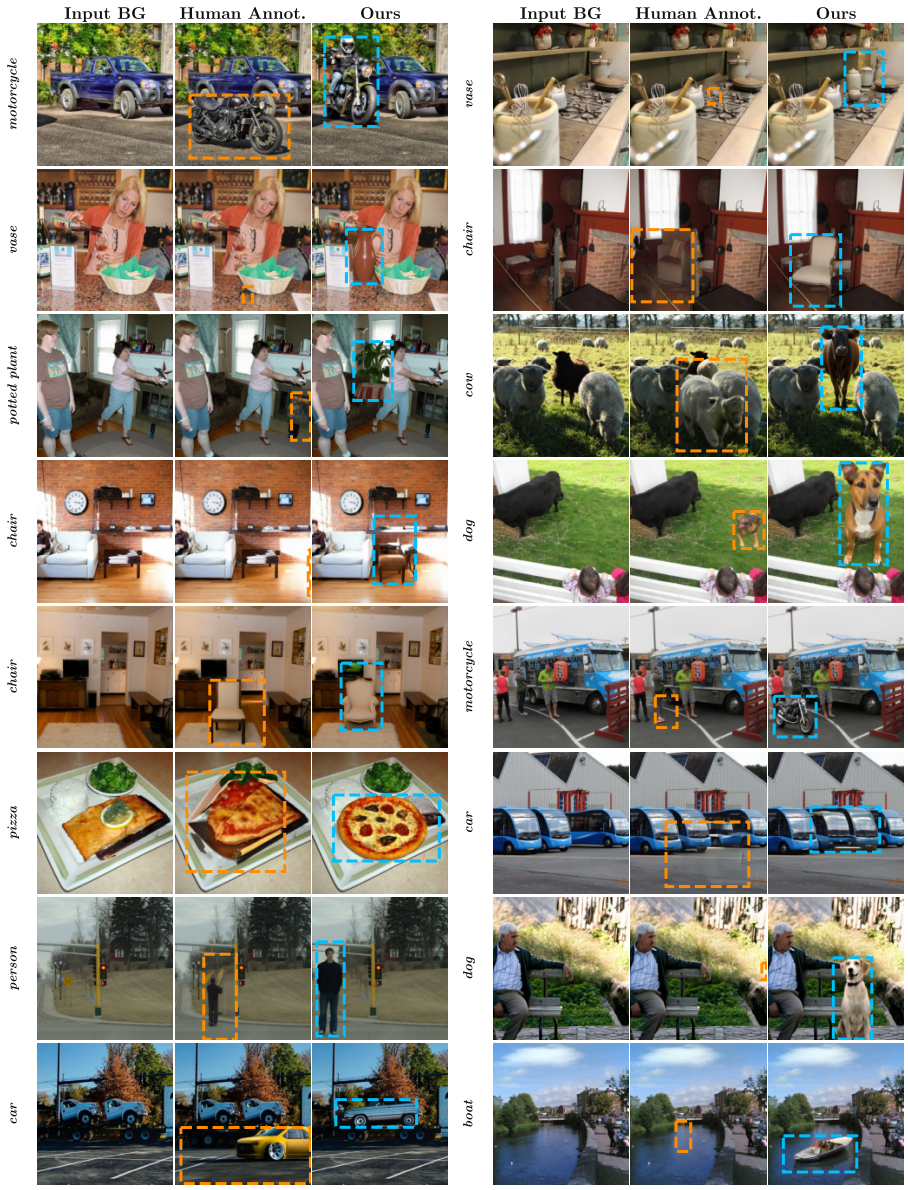


Fig. 17: Inpainting with Object Placement Priors. We evaluate three object placement approaches on the OPA backgrounds. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random).

Fig. 18: Inpainting with Object Placement Priors. We evaluate three object placement approaches on the OPA backgrounds. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random).

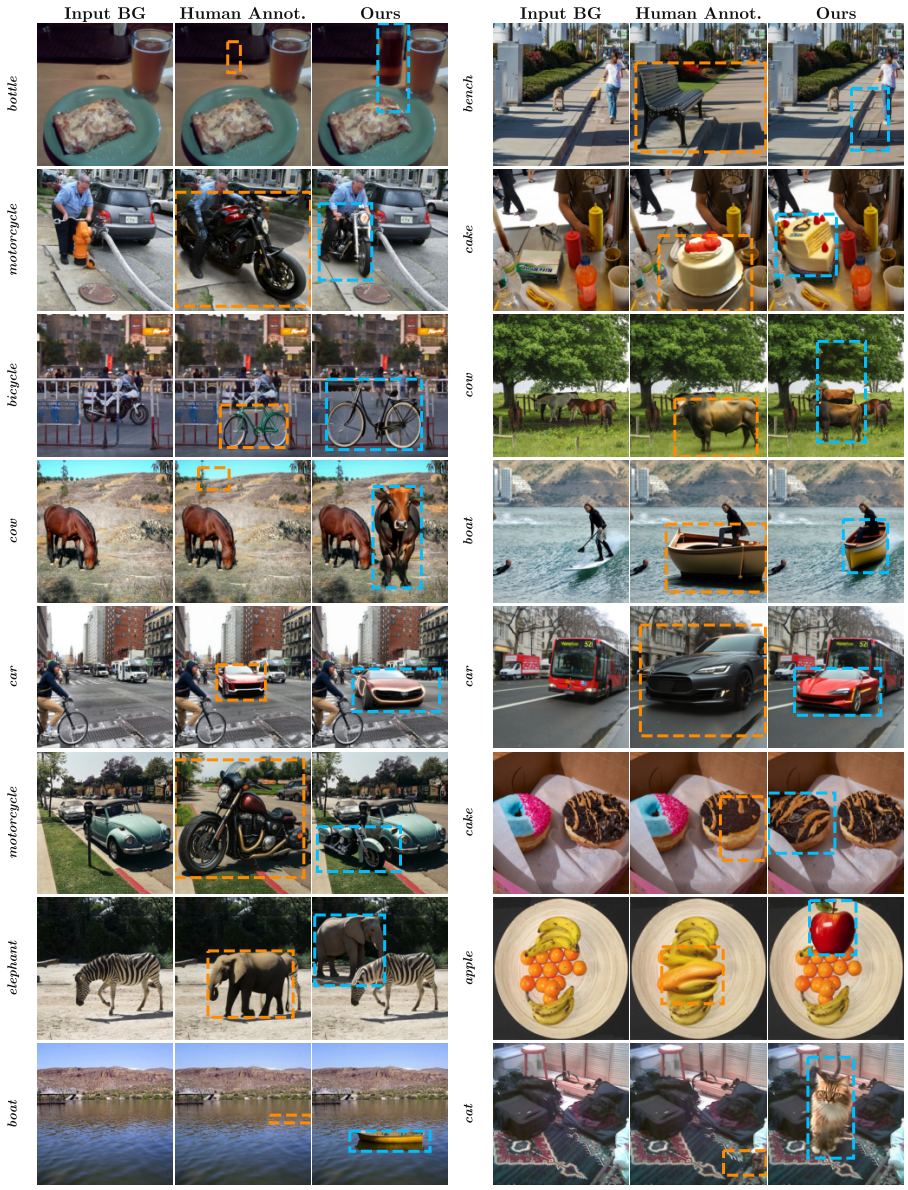


Fig. 19: Inpainting with Object Placement Priors. We evaluate three object placement approaches on the OPA backgrounds. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random).

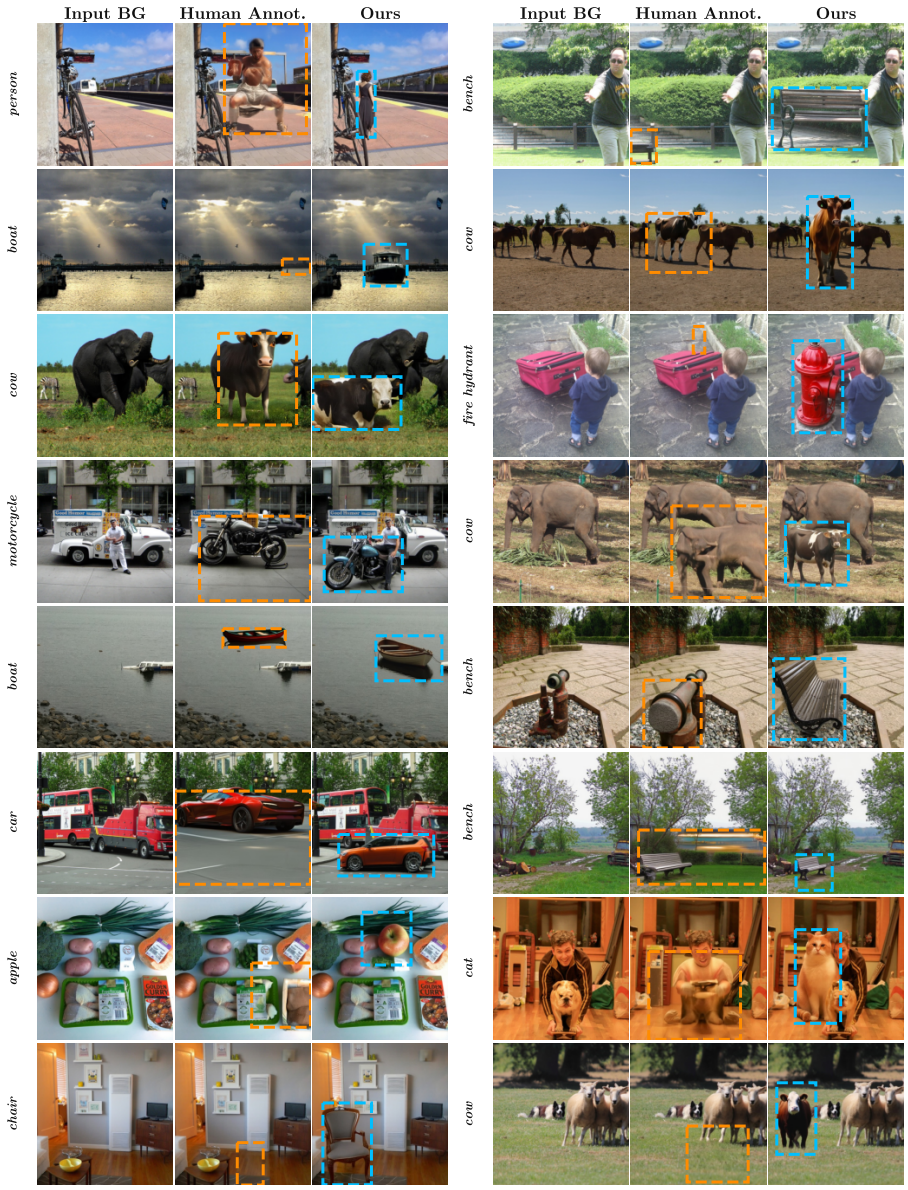


Fig. 20: Inpainting with Object Placement Priors. We evaluate three object placement approaches on the OPA backgrounds. *Human Annot.* uses ground-truths from OPA. *Random BBox* samples a uniformly random placement region, serving as a controlled baseline. *Ours* uses the top-ranked box of our dataset. Dashed boxes show the inpainting region (HiddenObject annotation (ours), OPA annotation, Random).