CatalogStitch: Dimension-Aware and Occlusion-Preserving Object Compositing for Catalog Image Generation

Sanyam Jain[1] Pragya Kandari[1] Manit Singhal[1] He Zhang[1] Soo Ye Kim[1]
[1]Adobe
{sanyjain, pkandari, manits, hezhan, sooyek}@adobe.com

Abstract

*Generative object compositing methods have shown remarkable ability to seamlessly insert objects into scenes. However, when applied to real-world catalog image generation, these methods require tedious manual intervention: users must carefully adjust masks when product dimensions differ, and painstakingly restore occluded elements post-generation. We present **CatalogStitch**, a set of model-agnostic techniques that automate these corrections, enabling user-friendly content creation. Our dimension-aware mask computation algorithm automatically adapts the target region to accommodate products with different dimensions; users simply provide a product image and background, without manual mask adjustments. Our occlusion-aware hybrid restoration method guarantees pixel-perfect preservation of occluding elements, eliminating post-editing workflows. We additionally introduce **CatalogStitch-Eval**, a 58-example benchmark covering aspect-ratio mismatch and occlusion-heavy catalog scenarios, together with supplementary PDF and HTML viewers. We evaluate our techniques with three state-of-the-art compositing models (ObjectStitch, OmniPaint, and InsertAnything), demonstrating consistent improvements across diverse catalog scenarios. By reducing manual intervention and automating tedious corrections, our approach transforms generative compositing into a practical, human-friendly tool for production catalog workflows. Our project page is at https://catalogstitch.github.io.*

1 Introduction

Product catalog imagery is essential for modern e-commerce and retail marketing. Creating high-quality catalog images traditionally requires expensive photoshoots, coordination of photographers and stylists, and weeks of production time. For businesses managing thousands of SKUs with seasonal updates, this process is prohibitively expensive and slow.

Figure 1. **Challenge 1: Product Dimension Mismatch.** When replacing a product with different proportions, freeform and bounding box masks distort the product shape. Our dimension-aware mask preserves correct proportions.

Recent advances in generative AI, particularly diffusion-based object compositing methods like ObjectStitch [8], Paint-by-Example [9], and AnyDoor [1], offer a promising foundation. These methods can seamlessly insert objects into scenes, handling geometry adjustment, color harmonization, and shadow generation in a unified framework. However, when we attempt to apply these methods to real-world catalog production workflows, critical gaps emerge that prevent practical deployment.

1.1 The Gap Between Research and Production

Existing compositing methods excel in controlled settings where: (1) replacement objects have similar proportions to originals, and (2) scenes are simple with no overlapping elements. Real-world catalog imagery violates both assumptions:

Challenge 1: Product Dimension Mismatch. When replacing one product with another, different proportions cause distortion when the new product is forced into the original target mask. For example, replacing a shoe in a scene with a differently proportioned shoe leads to shape distortion: the product is stretched or squashed to fit the original mask dimensions (Figure 1).

Challenge 2: Occlusion Destruction. Real catalog images feature products partially occluded by foreground elements. When the target region is replaced, these overlapping objects are destroyed or distorted by current methods. In Fig-

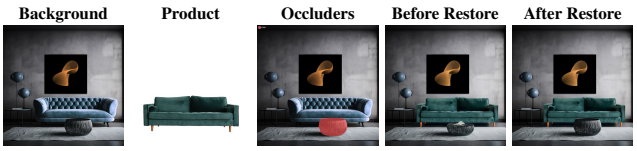

Figure 2. **Challenge 2: Occlusion Destruction.** Foreground elements (table/decor items) occluding the target product are corrupted during compositing. Our hybrid restoration preserves them pixel-perfectly.

ure 2, foreground furniture in front of the target sofa is corrupted during compositing.

1.2 Our Contributions

We present **CatalogStitch**, a collection of model-agnostic techniques that extend existing compositing methods for real-world catalog image generation:

1. **Dimension-Aware Mask Computation:** An algorithm that automatically computes an expanded target mask accommodating the new product's aspect ratio while staying centered on the original location. This preserves correct product proportions without manual intervention.
2. **Occlusion-Aware Hybrid Restoration:** A five-step approach that detects occluding objects, caches their exact pixels, applies generative inpainting to remove occluders and expose a clean background, performs generative compositing on that clean background, then restores the cached occluder pixels—guaranteeing pixel-perfect preservation of overlapping elements.
3. **CatalogStitch-Eval Benchmark:** A 58-example evaluation benchmark for catalog compositing, with 35 dimension-mismatch scenes, 23 occlusion scenes, masks, source metadata, PDF result summaries, and supplementary HTML viewers for rapid inspection. The benchmark package is publicly released[1].
4. **Cross-Model Evaluation:** An empirical study across three state-of-the-art compositing models (ObjectStitch, OmniPaint, InsertAnything) showing that the proposed preprocessing and postprocessing steps consistently improve realism and structural fidelity without changing the underlying generators.

Our key insight is that a *hybrid design*, combining generative AI for tasks such as harmonization and shadow generation with deterministic operations for tasks requiring exact fidelity, delivers both the flexibility of AI compositing and the reliability required for production workflows.

1.3 Enabling User-Friendly Compositing

Current compositing workflows demand significant manual effort: users must carefully craft masks when product dimensions differ, experiment with scaling parameters, and painstakingly restore occluded elements that get destroyed during generation. This technical overhead prevents non-expert users from leveraging generative compositing and slows down production pipelines even for experienced operators.

Our techniques shift the human-AI interaction from low-level pixel manipulation to high-level creative decisions. Users simply provide a product image and a background scene, while the system automatically handles mask adaptation and occlusion preservation. This *simplified interaction model* is crucial for democratizing AI-powered content creation and allowing marketers and designers to generate catalog imagery without specialized technical skills.

Furthermore, our model-agnostic design means these user-friendly capabilities can be added to any existing compositing model, future-proofing workflows as underlying generative models continue to improve.

2 Related Work

Generative Image Compositing. Recent diffusion-based methods have achieved impressive results in object compositing. ObjectStitch [8] uses a content adaptor to inject object features into a pre-trained inpainting model, handling geometry adjustment and harmonization jointly. Paint-by-Example [9] conditions image generation on reference images using CLIP embeddings. AnyDoor [1] employs ID tokens and high-frequency maps to preserve object identity during compositing. ControlCom [12] adds explicit control over object location and transformation. More recently, OmniPaint [10] and InsertAnything [7] have pushed the boundaries of compositing quality. Our work complements these methods by addressing preprocessing (mask computation) and postprocessing (occlusion restoration) challenges that are orthogonal to the core compositing model.

Object Placement and Harmonization. Prior work has studied where to place objects in scenes [14] and how to harmonize composited objects with backgrounds [2, 3]. The OPA dataset [4] provides benchmarks for evaluating object placement plausibility. Unlike these works that focus on placement selection or post-hoc harmonization, we address the mask adaptation problem that arises when the replacement object has fundamentally different dimensions than the original.

Amodal Completion and Occlusion. Amodal completion methods [5, 11] aim to predict the full extent of partially occluded objects. Layered image representations [13] decompose scenes into separate object layers. While related, our occlusion handling takes the opposite approach: rather than predicting occluded content, we *preserve* existing occluder pixels that would otherwise be destroyed during compositing. This deterministic approach guarantees fidelity that

[1] https://github.com/adobe-research/CatalogStitch

generative methods cannot match.

3 Method

We present two complementary techniques that can be applied as preprocessing and postprocessing steps around any generative compositing model. Our design philosophy prioritizes *user simplicity*: the user provides only a product image and a background scene, and our system automatically handles mask adaptation and occlusion preservation. Figure 3 summarizes the full end-to-end pipeline, while Figure 4 breaks down the two core modules.

3.1 Dimension-Aware Mask Computation

Problem Formulation. Given a background image I_b with a target region mask M_t indicating where to place the new product, and a product image I_p with product mask M_p, we need to compute an optimal target mask M_t^* that accommodates the product's proportions.

Let (w_t, h_t) be the bounding box dimensions of M_t and (w_p, h_p) be the dimensions of M_p. The aspect ratios are:

$$AR_t = \frac{w_t}{h_t}, \quad AR_p = \frac{w_p}{h_p} \tag{1}$$

When $|AR_t - AR_p| > \tau$ (we use $\tau = 0.06$), the product cannot fit naturally into the target region without distortion. Figure 4 (left) illustrates the decision process.

Algorithm. Our dimension-aware mask computation proceeds as follows:

1. **Extract Centroids:** Compute the centroid (c_x, c_y) of the original target region M_t to preserve spatial positioning.
2. **Compare Aspect Ratios:** If $|AR_t - AR_p| < \tau$, the original mask is sufficient; use $M_t^* = M_t$.
3. **Compute Optimal Dimensions:** When aspect ratios differ significantly:

$$h^* = h_t \tag{2}$$
$$w^* = h^* \cdot AR_p \tag{3}$$

   If $w^* < w_t$, we instead anchor on width:

$$w^* = w_t \tag{4}$$
$$h^* = w^*/AR_p \tag{5}$$

4. **Center on Original Location:**

$$x^* = \max(0, c_x - w^*/2) \tag{6}$$
$$y^* = \max(0, c_y - h^*/2) \tag{7}$$

5. **Generate Mask:** Create M_t^* as a rectangular mask with computed dimensions, clipped to image boundaries.

This ensures the dimension-aware mask expands to fit the new product's proportions while staying centered on the original target location, maintaining scene coherence.

3.2 Occlusion-Aware Hybrid Restoration via Exact Occluder Compositing

Problem Formulation. In professionally styled catalog images, products are often partially occluded by decorative elements (plants, vases, side tables). Let $\mathcal{O} = \{O_1, O_2, ..., O_k\}$ be the set of objects whose masks overlap with the target region M_t. When generative compositing replaces the target region, these occluders are typically destroyed or distorted.

Key Insight. Occluding objects are already observed in the original image with exact RGB values, scene-consistent illumination, shadows, and high-frequency structure. Instead of regenerating these regions, we cache their visible support and recompose them deterministically after generation. Figure 4 (right) summarizes the four restoration stages.

Five-Stage Hybrid Restoration Pipeline.

Step 1: Segment Overlapping Foreground Instances. Using entity segmentation on the background image I_b, we identify candidate foreground instances and their masks. For each detected entity E_i with bounding box B_i, we compute overlap with the target region:

$$\text{IoU}(B_i, B_t) = \frac{|B_i \cap B_t|}{|B_i \cup B_t|} \tag{8}$$

If IoU $> \tau_{occ}$ (we use $\tau_{occ} = 0.01$ to capture minimal overlaps), E_i is marked as an occluder.

Step 2: Cache Exact Occluder Support. For each occluding entity O_i:

- Extract the exact visible pixel support from the original image: $P_i = I_b \odot M_{O_i}$
- Cache $(P_i, M_{O_i}, \text{coords}_i)$ for deterministic reuse
- No filtering or synthesis is applied; the cached region is preserved verbatim

Step 3: Generative Inpainting of Occluder Regions. With occluder positions cached, we remove their visual interference from the background by applying generative inpainting within the union of occluder masks $\bigcup_i M_{O_i}$:

$$I_b^{\text{inp}} = \text{Inpaint}\Big(I_b, \bigcup\nolimits_i M_{O_i}\Big) \tag{9}$$

This yields a clean background I_b^{inp} in which the target region is fully unoccluded, allowing the compositor to integrate the replacement product without residual occluder interference.

Step 4: Run Conditional Generative Compositing. Apply the compositing model (ObjectStitch, OmniPaint, or InsertAnything) using the inpainted background and the dimension-aware mask M_t^* from Section 3.1:

$$I_{comp} = \text{Composite}(I_b^{\text{inp}}, I_p, M_t^*, M_p) \tag{10}$$

Because occluders are absent from I_b^{inp}, the compositor generates a clean product integration with no occluder artefacts.

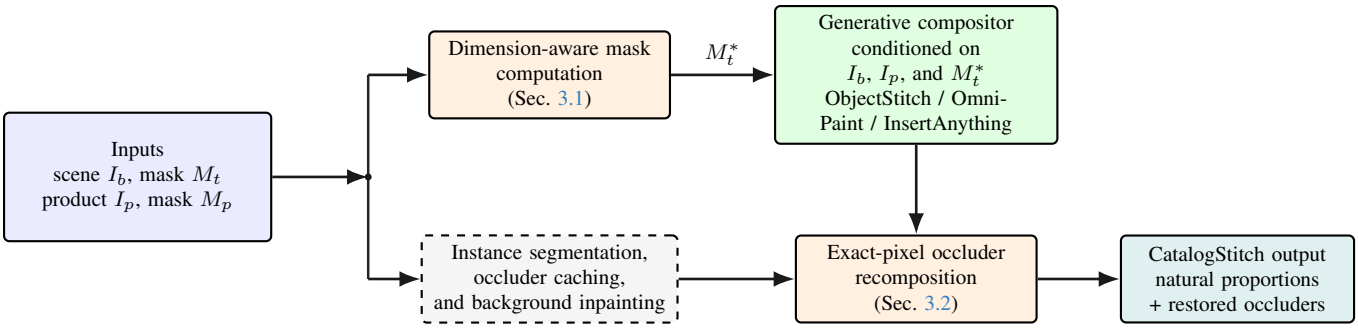

Figure 3. **CatalogStitch overview.** Two lightweight, model-agnostic modules wrap a baseline compositor. The target mask is adapted to the replacement product ratio; occluders are segmented, cached, and inpainted away before compositing, then restored from the original pixels at the final step.

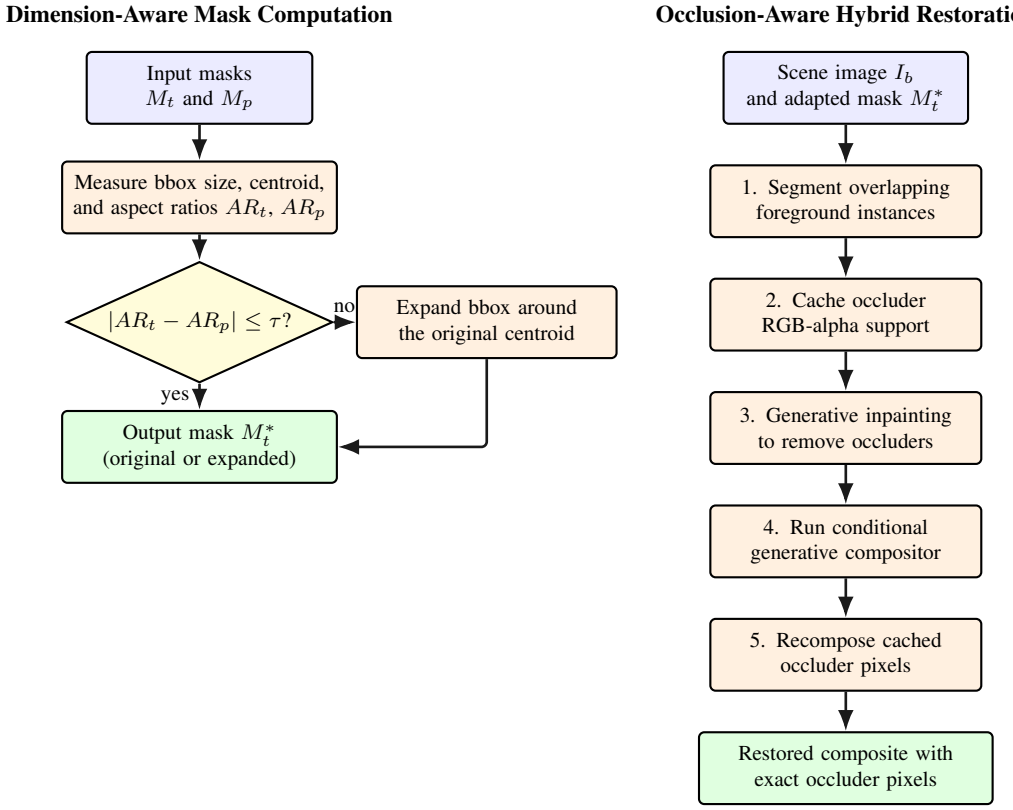

Figure 4. **Module-level flow charts.** Left: dimension-aware mask computation preserves the original placement when the mismatch is small and otherwise expands the target region around the original centroid. Right: occlusion-aware restoration detects overlapping entities, caches their exact pixels, applies generative inpainting to remove occluders and expose a clean background, runs the compositor on the clean background, and finally restores the cached occluder pixels to preserve original geometry, texture, and scene-consistent lighting.

Step 5: Recompose Cached Occluders. Alpha-composite the cached occluder regions onto the generated result:

$$I_{final} = I_{comp} \odot (1 - \bigcup_i M_{O_i}) + \sum_i P_i \qquad (11)$$

The reinserted occluders preserve their original geometry, texture, lighting, and shadows by construction.

4 Experiments

4.1 Implementation Details

All techniques are purely inference-time wrappers requiring no model fine-tuning or additional training. The dimension-aware mask computation is a closed-form geometric calculation; occlusion detection uses a single forward pass of EntitySeg [6]; recomposition uses pixel-level alpha blending. The combined overhead is negligible relative to the genera-

tive compositing step. All experiments use images at native resolution (typically 1024×1024 or higher). The threshold τ=0.06 controls when dimension-aware mask computation is applied, balancing unnecessary mask expansion against missed adaptation cases. The IoU threshold τ_{occ}=0.01 controls occluder detection sensitivity, capturing minimal occlusions without including non-overlapping nearby objects.

4.2 Experimental Setup

Evaluation Dataset. We curated **CatalogStitch-Eval**, a challenging evaluation dataset for catalog image compositing consisting of two subsets:

- **Dimension Mismatch Set:** 35 image pairs where the replacement product has significantly different aspect ratio than the target region (e.g., tall lamp replacing wide table, square bag replacing rectangular clutch).
- **Occlusion Set:** 23 images featuring products partially occluded by 1–2 foreground elements (plants, vases, side tables, lamps).

The dataset package (metadata, pre-computed masks, method outputs, and HTML viewers) is publicly released[2]. Full qualitative results for all 35 dimension-aware and 23 occlusion examples are provided in the supplementary material (`additional_results_dimension_aware.pdf` and `additional_results_occlusion.pdf`).

Baseline Methods. We evaluate our techniques as preprocessing/postprocessing steps applied to three state-of-the-art compositing models:

- **ObjectStitch** [8]: Content adaptor-based approach for reference-guided inpainting.
- **OmniPaint** [10]: Unified insertion-removal inpainting framework with disentangled editing.
- **InsertAnything** [7]: In-context editing approach for reference-based object insertion via DiT.

For each baseline, we compare: (1) the original method with standard masks, and (2) the method enhanced with our CatalogStitch techniques.

4.3 Qualitative Results

Dimension-Aware Mask Results. Figure 5 shows representative high-mismatch examples across OmniPaint, InsertAnything, and ObjectStitch. The supplementary material contains all 35 benchmark cases for all three compositors.

Key observations:

- Without dimension-aware mask adaptation, products are stretched, squashed, or cropped to fit the original target region.
- With the adapted mask, products retain their native proportions while staying aligned with the original scene layout.
- The same mask computation generalizes across all three baseline compositors, supporting a model-agnostic interface improvement rather than a model-specific fix.

Occlusion Handling Results. Figure 6 shows representative occlusion-heavy scenes where foreground objects are visibly restored after compositing. The supplementary material provides before/after results for all 23 benchmark examples.

Key observations:

- Without occlusion handling, foreground elements are destroyed, distorted, or inconsistently hallucinated.
- With hybrid restoration, overlapping objects recover their exact geometry, texture, and illumination because the original pixels are pasted back.
- The benefit is most visible for fine structures and hard boundaries that diffusion-based models struggle to reconstruct reliably.

4.4 Quantitative Evaluation

Table 1 reports quantitative results for the six method variants. Baselines use standard bbox masks; "+ Ours" uses dimension-aware masks (and for occlusion, exact-pixel restoration). We evaluate using five metrics: **AR Error** measures aspect-ratio preservation of the inserted product (lower is better). **FID** (Fréchet Inception Distance) captures the distributional similarity between input-product and generated-object crops—lower values indicate more realistic composites. **CLIP-score** measures semantic alignment between the composited region and the reference product using CLIP embeddings (higher is better). **DINO-score** evaluates structural similarity via self-supervised DINO features, capturing fine-grained appearance fidelity (higher is better). **Occluder PSNR** measures pixel-level restoration quality over occluder regions (higher is better).

Across all baselines, adding our proposed components consistently improves every metric: AR Error drops from ∼30% to ∼4–5%, FID decreases substantially, and both CLIP and DINO scores increase. Among all configurations, **InsertAnything + Ours achieves the best results across all five metrics**, reaching the lowest FID (77.72), highest CLIP-score (92.68), highest DINO-score (88.30), lowest AR Error (3.92), and highest Occluder PSNR (27.54).

Aspect Ratio Preservation. For the 35 dimension-mismatch examples, we measure:

$$\text{AR Error} = \frac{|AR_{output} - AR_{input}|}{AR_{input}} \times 100\% \qquad (12)$$

where AR_{output} is measured from the generated object region and AR_{input} from the input product.

[2] https://github.com/adobe-research/CatalogStitch

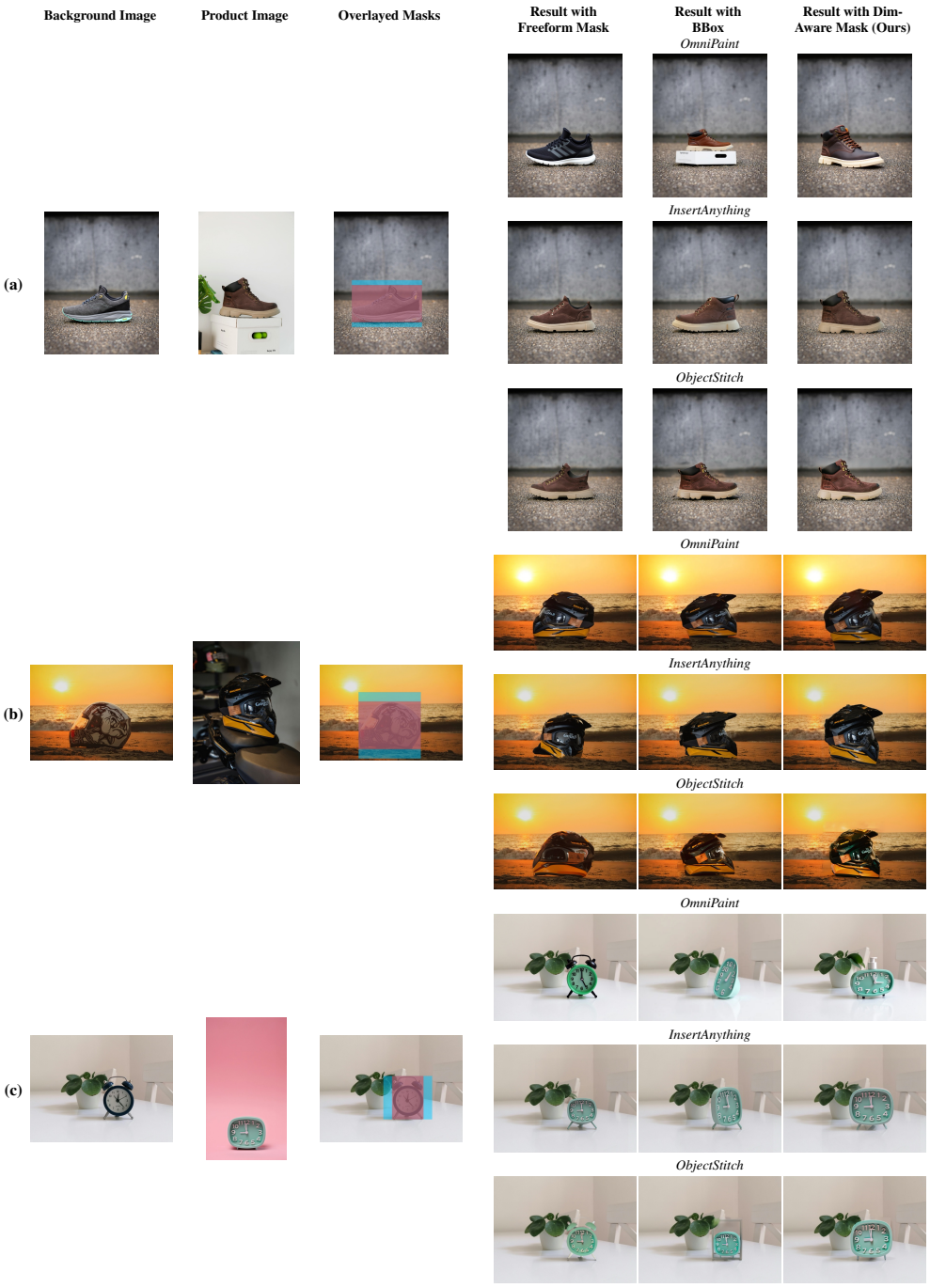

Figure 5. **Dimension-aware mask computation across models.** Left inputs: background and product. Mask overlay visualizes the dim-aware (blue) and bounding-box (pink) regions. Right: outputs under Freeform, BBox, and Dim-Aware masks for OmniPaint, InsertAnything, and ObjectStitch. Dim-aware masks preserve object proportions more reliably than Freeform and BBox masks across all models.

Figure 6. **Occlusion-aware hybrid restoration across models.** Left inputs: background and product. Mask overlay and occluder composite are visualizations of the occlusion regions. Right: Freeform, BBox, and Dim-Aware outcomes for InsertAnything and ObjectStitch, shown before and after occluder restoration. Our pipeline segments occluders, caches their pixels, inpaints them away to expose a clean background for compositing, and finally pastes the cached pixels back—preserving overlapping structures more faithfully than the corresponding before-restoration outputs.

Method	AR Error ↓	Occluder PSNR ↑	FID ↓	CLIP-score ↑	DINO-score ↑
OmniPaint	31.07	-	137.15	83.03	68.50
OmniPaint + Ours	4.57	-	135.79	86.11	72.99
ObjectStitch	30.97	11.60	101.55	90.27	85.76
ObjectStitch + Ours	5.05	26.84	91.52	90.62	88.09
InsertAnything	29.98	13.33	105.99	90.23	82.63
InsertAnything + Ours	**3.92**	**27.54**	**77.72**	**92.68**	**88.30**

Table 1. **Quantitative comparison on CatalogStitch-Eval.** AR Error, CLIP, and DINO scores are computed on the 35 dimension examples; Occluder PSNR on the 23 occlusion examples (– for OmniPaint). CLIP and DINO are cosine similarity $\times 100$; FID is between input-product and generated-object crop distributions.

Occluder Fidelity. For the 23 occlusion examples (InsertAnything/ObjectStitch), we report masked PSNR over restoration regions estimated by the before/after delta; higher values indicate stronger recovery toward the source scene.

4.5 Discussion

Cross-Model Consistency. Our techniques yield consistent improvements across all compositor architectures. AR Error drops from $\sim$30% to $\sim$4–5% uniformly across OmniPaint, InsertAnything, and ObjectStitch, confirming that dimension mismatch and occluder destruction are *systematic* failure modes shared across generative pipelines rather than artifacts of any single architecture.

Complementarity of Techniques. The dimension-aware and occlusion-aware modules address orthogonal failure modes and compose naturally. In the 23 occlusion examples, many also exhibit dimension mismatch; applying both modules in sequence yields the strongest results across every metric (Table 1).

Failure Cases. When aspect-ratio mismatch is very large, the expanded mask may cover important scene context, degrading background coherence. For occlusion restoration, translucent occluders may introduce subtle compositing seams at alpha boundaries. Under-segmentation of closely packed objects can also leave restore regions incomplete.

5 Conclusion

We presented CatalogStitch, a set of model-agnostic techniques that bridge the gap between research-grade object compositing methods and production catalog image generation requirements. Our dimension-aware mask computation algorithm addresses the dimension mismatch problem by automatically adapting target regions to accommodate products with different aspect ratios. Our occlusion-aware hybrid restoration method guarantees pixel-perfect preservation of foreground elements that would otherwise be destroyed during compositing.

By automating tedious manual corrections, our techniques transform the human-AI interaction model for compositing: users focus on high-level creative decisions (which product, which background) while the system handles low-level technical adjustments. This simplification enables non-expert users to leverage powerful generative models without specialized skills.

We demonstrated our techniques across three state-of-the-art compositing models (ObjectStitch, OmniPaint, InsertAnything), showing consistent improvements on challenging catalog scenarios involving dimension mismatches and occlusions.

A central insight of our work is that *hybrid approaches*, combining generative AI for harmonization and shadow generation with deterministic methods for mask computation and pixel restoration, achieve both the flexibility of AI compositing and the reliability required for production quality. Fully generative approaches cannot guarantee the fidelity needed for professional catalog imagery.

Limitations and Future Work. Our current approach assumes occluders are in front of the target product; handling complex multi-layer occlusion relationships is left for future work. The dimension-aware mask computation uses rectangular bounding boxes; exploring shape-aware mask adaptation could further improve results. Additionally, broader quantitative and human-preference evaluation across larger datasets and more compositors would strengthen the empirical validation.

References

[1] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 1, 2

[2] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8394–8403, 2020. 2

[3] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18470–18479, 2022. 2

[4] Liu Liu, Zhenchen Liu, Bo Zhang, Jiangtong Li, Li Niu, Qingyang Liu, and Liqing Zhang. Opa: Object placement assessment dataset. *arXiv preprint arXiv:2107.01889*, 2021. 2

[5] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. Pix2gestalt: Amodal segmentation by synthesizing wholes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2024. 2

[6] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Zhe Lin, Philip Torr, and Jiaya Jia. Open world en-

tity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8743–8756, 2023. 4

[7] Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in dit. *arXiv preprint arXiv:2504.15009*, 2025. 2, 5

[8] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023. 1, 2, 5

[9] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 1, 2

[10] Yongsheng Yu, Ziyun Zeng, Haitian Zheng, and Jiebo Luo. Omnipaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. *arXiv preprint arXiv:2503.08677*, 2025. 2, 5

[11] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene deocclusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3792, 2020. 2

[12] Bo Zhang, Yuxuan Duan, Jun Lan, Yan Hong, Huijia Zhu, Weiqiang Wang, and Li Niu. Controlcom: Controllable image composition using diffusion model. *arXiv preprint arXiv:2308.10040*, 2023. 2

[13] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 2

[14] Lingzhi Zhang, Tarmily Wen, Jianbo Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *European Conference on Computer Vision*, pages 566–581, 2020. 2