Visual Jenga: Discovering Object Dependencies via Counterfactual Inpainting

Anand Bhattad[1] Konpat Preechakul[2] Alexei A. Efros[2]
[1]Toyota Technological Institute at Chicago [2]University of California, Berkeley
https://visualjenga.github.io

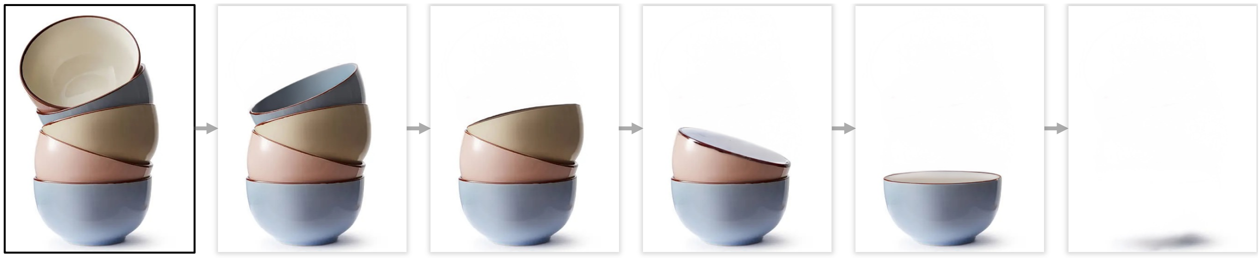


Figure 1. **Visual Jenga:** Given an input image (left), we generate a sequence of images, removing one object at a time while keeping the scene stable. We argue that this new task provides a useful signal to assess the level of grounded scene understanding of vision systems. *See the video on our project page for animated results.*

Abstract

This paper proposes a novel scene understanding task called Visual Jenga. Drawing inspiration from the game Jenga, the proposed task involves progressively removing objects from a single image until only the background remains. Just as Jenga players must understand structural dependencies to maintain tower stability, our task reveals the intrinsic relationships between scene elements by systematically exploring which objects can be removed while preserving scene coherence in both physical and geometric sense. As a starting point for tackling the Visual Jenga task, we propose a simple, data-driven, training-free approach that is surprisingly effective on a range of real-world images. The principle behind our approach is to utilize the asymmetry in the pairwise relationships between objects within a scene and employ a large inpainting model to generate a set of counterfactuals to quantify the asymmetry.

1. Introduction

Can one truly understand a scene by simply naming the objects in it? While modern computer vision methods excel at object detection and semantic segmentation, these capabilities often prove inadequate for practical purposes, such as vision-guided robot manipulation or truly grounded image editing. Treating scenes as static collections of isolated elements, recognition models neglect the critical relationships between objects that give scenes their intrinsic meaning. In this paper, we argue that true *scene understanding* necessitates understanding how objects depend on and interact with one another within the space of a scene.

Drawing inspiration from the game Jenga[1], we propose a novel task: to virtually deconstruct scenes by carefully erasing objects, much like players strategically remove blocks from a tower while making sure it does not collapse. As shown in Fig. 1, the goal of Visual Jenga is to progressively remove objects from a single image, one at a time, such that the scene always remains "well-formed". Solving this task reveals the relationships between scene elements by systematically exploring which objects can be removed while preserving scene coherence both physically and geometrically.

By framing scene understanding as a sequential deconstruction task, Visual Jenga allows us to evaluate how objects relate to and depend on one another: an aspect central to scene understanding in humans [8] yet largely overlooked by current benchmarks. Such understanding is crucial in many practical domains. For instance, the ability to remove objects without destabilizing a scene is essential for many robotic manipulation tasks [17, 39]. Preserving physical scene coherence is also important for realistic image editing applications.

As a starting point for solving the Visual Jenga task, we propose a simple, data-driven approach that is surprisingly effective in a range of real-world scenes without requiring any explicit physical reasoning. Our approach uses a form of counterfactual reasoning by asking "what if this object were removed?" The principle behind our approach is using

---

[1] Jenga is derived from the Swahili word "kujenga" meaning "to build". There are similar games in other cultures: pick-up sticks, mikado, jonchets, бирюльки, etc.

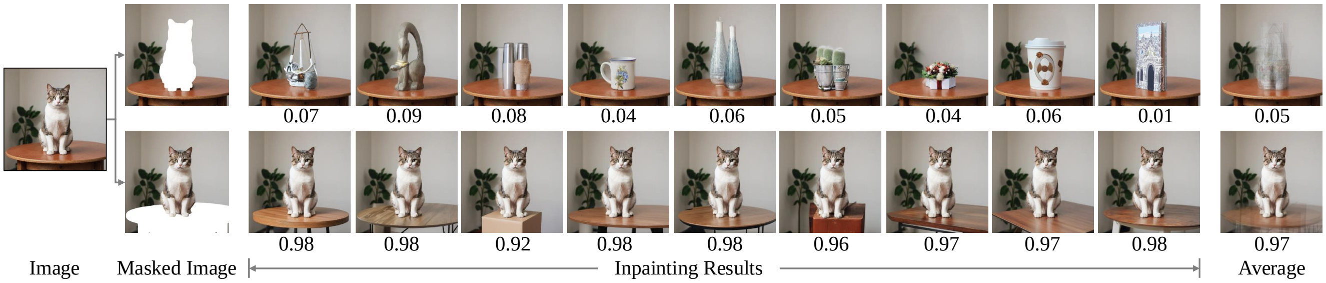


Figure 2. **Counterfactual Inpainting.** Given a pair of objects in an image (here, cat and table), we want to compute which of the two objects is more dependent on the other. We do this by removing each object in turn (masked images), and use a large inpainting model to generate N possible inpaintings for the masked regions (top and bottom rows). The number below each inpainting result is a pairwise cosine similarity (derived from CLIP and DINO) between it and the object in the original image. The average images (for illustration only) show that the cat could be replaced by many different objects, while the table remains largely unchanged. This suggests that the table is supporting the cat.

the *asymmetry* in the pairwise relationships between objects within a scene [43].

Consider the cat sitting on a table in Fig. 2. The cat depends on the table below for structural support, but not vice versa. If we consider the counterfactual of removing the table from the image, the cat would require some other support surface for the scene to remain stable [8]. If, on the other hand, we remove the cat, the scene is already stable, so it doesn't much matter what, if anything, will go in the cat's place. To make this intuition quantifiable, we use an off-the-shelf large image inpainting model to help us estimate the conditional probabilities of counterfactual images. This does not require any training and exploits existing knowledge of what constitutes a well-formed scene [8] already captured in large generative models. As Fig. 2 shows, inpainting the region occupied by the cat results in a diverse range of plausible objects that could replace it, whereas inpainting the table consistently produces similar support structures for the cat. Averaging these differences over multiple inpainting passes allows us to quantify this asymmetry and determine which object should be removed first.

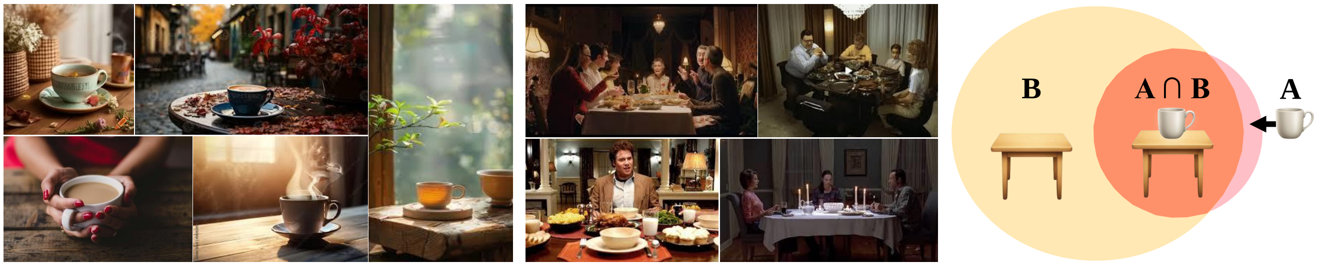
In summary, our contributions are: (1) Visual Jenga task: a novel scene understanding task that evaluates object dependencies through sequential removal, inspired by counterfactual reasoning. (2) Counterfactual Inpainting approach: a training-free method that quantifies object dependencies, exploiting asymmetry in object co-occurrences using large-scale generative inpainting models. We demonstrate our approach through a quantitative pairwise evaluation as well as qualitative, full-scene decompositions.

2. Related work

At the dawn of computer vision in the 1960s, when perception and action were considered two sides of the same coin, the grand goal of image understanding was the ability to reason about the physical scene from an image. Roberts' BlocksWorld [56], the very first PhD thesis in computer vision, was all about analyzing object relationships within a physical scene (made up of simple blocks) so that a robot could pick up these blocks one-by-one and reassemble them into a different configuration. Alas, in the 60 years that followed, the goal of image understanding has been watered down to a combination of object detection and semantic "segmentation" [35], and nowadays, image captioning. In this section, we will review prior work that considered the task of scene understanding in its original meaning.

Qualitative 3D scene understanding. Psychologist Irving Biederman's classic work on scene perception [8] argues that the way humans interpret visual scenes goes far beyond a list of objects or a text description. Biederman identified several physical and geometric relational constraints between scene objects (such as physical support and occlusion) that must be satisfied for a scene to be well-formed. Inspired by this, Hoiem et al. focused on incorporating Biederman's constraints into their scene understanding systems [24, 29, 30, 59]. Subsequent research built on this by developing layered scene representations via layer-wise decomposition [32, 68], object-level deocclusion [42], and multi-layer reasoning [15]. More recently, physics-aware scene understanding inspired by the original BlocksWorld [56] has been revisited, both in synthetic settings [37, 39], as well as in attempts to generalize it to real-world scenes [25, 45, 60].

Counterfactual spatial reasoning. Identifying causal relationships in the real world from observation has been an open problem in the causal inference community [46]. In computer vision, Lopez-Paz et al. [43] have considered the special case of an object causing the presence of another object using *causal disposition* as a measure. Goyal et al. [22] use visual interventions to explain model decisions by showing how modifying specific image elements alters predictions. Besserve et al. [5] introduced counterfactual interventions to pinpoint modular components in generative networks, enabling targeted image editing and causal anal-



Search: "Cup" Search: "Table" $\text{P(Table|Cup)} \gg \text{P(Cup|Table)}$

Figure 3. **Asymmetric Relationships in Real-World Images.** Consider performing two internet image searches: "cup" (left) and "table" (middle). Notice that almost all the cups are depicted on top of a table, whereas images of tables rarely contain cups. That is, $P(\text{Table} \mid \text{Cup}) \gg P(\text{Cup} \mid \text{Table})$. The Venn diagram (right) illustrates how object A (cup) depends on object B (table) for structural support: observing a table does not guarantee a cup, but observing a cup strongly implies a table (i.e. $P(\text{Table} \mid \text{Cup}) \gg P(\text{Cup} \mid \text{Table})$). By leveraging these asymmetric relationships, we can infer object dependencies in a scene from the distributions $P(A \mid B)$ and $P(B \mid A)$ learned from large-scale data.

ysis of internal representations. Zhou et al. [69] evaluates human responses to synthetic block tower simulations to understand how people assess physical support relations. Our work scales up the principles laid by these works to complex real-world data using large pretrained generative models. While observational data may only provide statistical co-occurrence information, which is not truly causal, large-scale models trained on images [2, 26] and videos [4] show impressive counterfactual modeling capabilities, via text prompting [9], visual prompting [3], and even simple classification [38]. The underlying visual understanding of a well-formed scene in generative models covers a wide range of attributes, like geometry, materials, lighting and support, among others [6, 7, 16, 67], and has shown promise for identifying object segmentation [48] and even amodal segmentation [50].

Object dependencies and scene graphs. Another attempt at deeper scene understanding is a line of work representing a scene as a graph of relations between atomic units, such as objects. Visual Memex [44], an early work in this area, treated each object as a node with edges for visual similarity, spatial co-occurrences, etc. Visual Genome [36] extended this to object categories using a large-scale crowd-sourced scene graph, spurring further research [63, 66]. However, these methods focus on 2D relationships and neglect geometric or physical aspects. Other efforts factorize images into object-centric latent components [11, 18, 19, 23, 51], but they treat objects independently, missing interactions and support relations.

Object removal. Prior work has largely focused on evaluating the visual quality of object removal, insertion, and inpainting [12, 61, 70]. Existing benchmarks assess inpainting quality using standard datasets [14, 40, 41, 65] and measure object removal performance [47, 58]. However, to our knowledge, no benchmark explicitly evaluates structural dependencies between objects or validates the correctness of object removal sequences of all objects in the scene.

3. Visual Jenga task

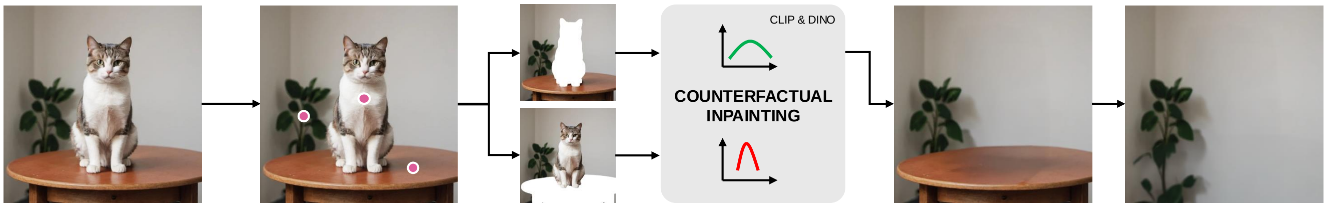
Visual Jenga task aims to evaluate scene understanding capabilities beyond passive visual observation, pushing towards physical object interaction understanding [20, 31]. Given a single input image, an algorithm needs to simulate an "action on the scene" by generating a sequence of images where it removes one object at a time until only the background remains while maintaining scene coherency and stability (Fig. 1). Successfully removing objects without destabilizing the scene demonstrates an understanding of object dependencies. We next introduce a simple, training-free approach to the Visual Jenga task that infers removal order based on object co-occurrence, without relying on any explicit physical reasoning.

3.1. Dependency as conditional probabilities

Consider the illustrative example on Fig. 3: performing an Internet image search for "cup" returns many images featuring tables, while a search for "table" rarely shows cups. This fundamental asymmetry reveals the dependencies between objects in a scene, and has been used to uncover object causal connections [43]. Let $A = \text{cup}$ and $B = \text{table}$. Shown as a Venn diagram in Fig. 3, the asymmetric relationship can be captured by

$$\begin{cases} P(A \mid B) \ll P(B \mid A) & \Rightarrow \text{A depends on B} \\ P(A \mid B) \gg P(B \mid A) & \Rightarrow \text{B depends on A} \end{cases} \tag{1}$$

In our example, $P(A \mid B)$ is very small (i.e., $P(\neg A \mid B)$ is large), while $P(B \mid A)$ is very large (i.e., $P(\neg B \mid A)$ is small). This asymmetric relationship uncovers not only the existence but also the direction of dependency, extending Reichenbach's principle of the common cause [21, 55]. Our approach, which compares $P(A \mid B)$ and $P(B \mid A)$, is more practical than the probabilistic theory of causation [27], that compares $P(A \mid B)$ with $P(A \mid \neg B)$, because the latter requires counterfactual reasoning about the removal of B, demanding information beyond object co-occurrences.



Input Image a) Molmo [13] b) SAM2 [54] c) See Figure 2 d) Removal with *Firefly* [1]

Figure 4. **Our Pipeline.** Given only an input image, (a) we first run Molmo [13] which places a point on each object in the image. (b) These points then serve as prompts for the Segment-Anything (SAM 2) [54] model to obtain segmentation maps for each object. (c) Given the object masks, we can now run our Counterfactual Inpainting method on all object candidates to determine their removal order via a ranking strategy (illustrated in Fig. 2). (d) Finally, we use Firefly [1] to remove objects based on these ranking order.

3.2. Counterfactual inpainting method

Generalizing to a real scene with more than two objects, the conditional probability $P(A|B)$ is replaced by the complete notation $P(A|B, \text{rest})$, where "rest" denotes the remainder of the scene. To estimate this, we start from the fact that large generative models capture the distribution $P(x)$, where x is an image that may contain objects of any kind. For a particular image $x = X$ that contains objects A and B, we can approximate $P(A|B, \text{rest})$ with $P(A|X - A)$ by masking out object A and using a large generative model to inpaint (hole-fill) the region corresponding the mask, given the rest of the image. Similarly, we obtain $P(B|X - B)$ for object B. By comparing these two quantities according to the rules in Eq. (1) as illustrated in Fig. 2, we can infer the object dependencies purely from co-occurrence statistics.

3.3. Practical details

The above algorithm is simple and principled, but to make it practical, we need to specify how to: 1) obtain object masks, 2) reliably compute conditional probabilities of inpainted images, and 3) choose which object to remove first. We describe our choices below and illustrate them in Fig. 4. None of these choices should be considered definitive. We expect them to change as technology matures.

1. Obtaining the object mask. To get object masks in the scene, we use off-the-shelf models. We first extract object coordinates using MOLMO [13] (Fig. 4a), and then use these as prompts for SAM 2 [54] to obtain segmentation maps without class labels (Fig. 4b).

2. Obtaining reliable conditional probability. Directly extracting likelihoods from image diffusion models is unreliable for two reasons: first, $P(A \mid X - A)$ for a specific object A is noisy; second, diffusion models are not optimized for likelihood scoring [28, 34]. Therefore, rather than focusing on a specific instance of A, we consider a semantic class of A and evaluate the "peakedness" of the distribution $P(A \mid X - A)$. We call this measure the **diversity** score of A (Fig. 4c). To compute it, we first gather N different inpaintings of A, denoted c^j_{new} for $j \in [1, N]$, using Runway's checkpoint of Stable Diffusion 1.5 [57]. We then quantify how semantically diverse these N inpaintings are using both CLIP [53] and DINO [49] features.

$$1 - \frac{1}{N}\sum_{j=1}^{N} \text{CLIPSIM}(c^j_{\text{new}}, c_{\text{orig}}) \times \frac{1}{N}\sum_{j=1}^{N} \text{DINOSIM}(c^j_{\text{new}}, c_{\text{orig}}) \tag{2}$$

where CLIPSIM and DINOSIM are cosine similarity over CLIP and DINO representations (normalized to $[0, 1]$ and normalized by the segmentation area fraction of the crop), c_{orig} is the original crop. Using either CLIP or DINO alone works, but having both is more robust (see Table 2).

3. Removing the most "diverse" object. After computing the diversity scores for all objects in the image, we remove the object with the *highest* diversity score first using an off-the-shelf object remover, Adobe Firefly [1] (Fig. 4d). If a new object appears (for example because of occlusion) after an object removal is also included in the ranking by rerunning the whole pipeline again.

4. Evaluation

Visual Jenga outputs a sequence of images, making human visual inspection a natural evaluation method. To complement qualitative assessment, we also perform automatic quantitative evaluation (all eval data provided on the project webpage). Our evaluation comprises three parts: pair-wise object ordering (Sec. 4.1), complete scene decomposition (Sec. 4.2), and comparison to simple heuristics (Sec. 4.3).

4.1. Pair-wise object ordering

To assess object dependency ordering in an automated manner, we test the model's ability to determine which of two given objects (specified by masks) should be removed first when physical constraints dictate a clear order. We created two test sets:

NYU-v2: We use NYU Depth V2 dataset [59], which contains 1449 RGBD images of indoor scenes. Using support relation annotations from Yang et al. [64], we extracted 485 unique images yielding 668 pair-wise comparisons with unambiguous removal ordering (details in Supp.).

HardParse: Because NYU-v2 dataset has very few examples of complex object depencencies (e.g. stacks of objects, hanging/leaning objects), we also produced a more

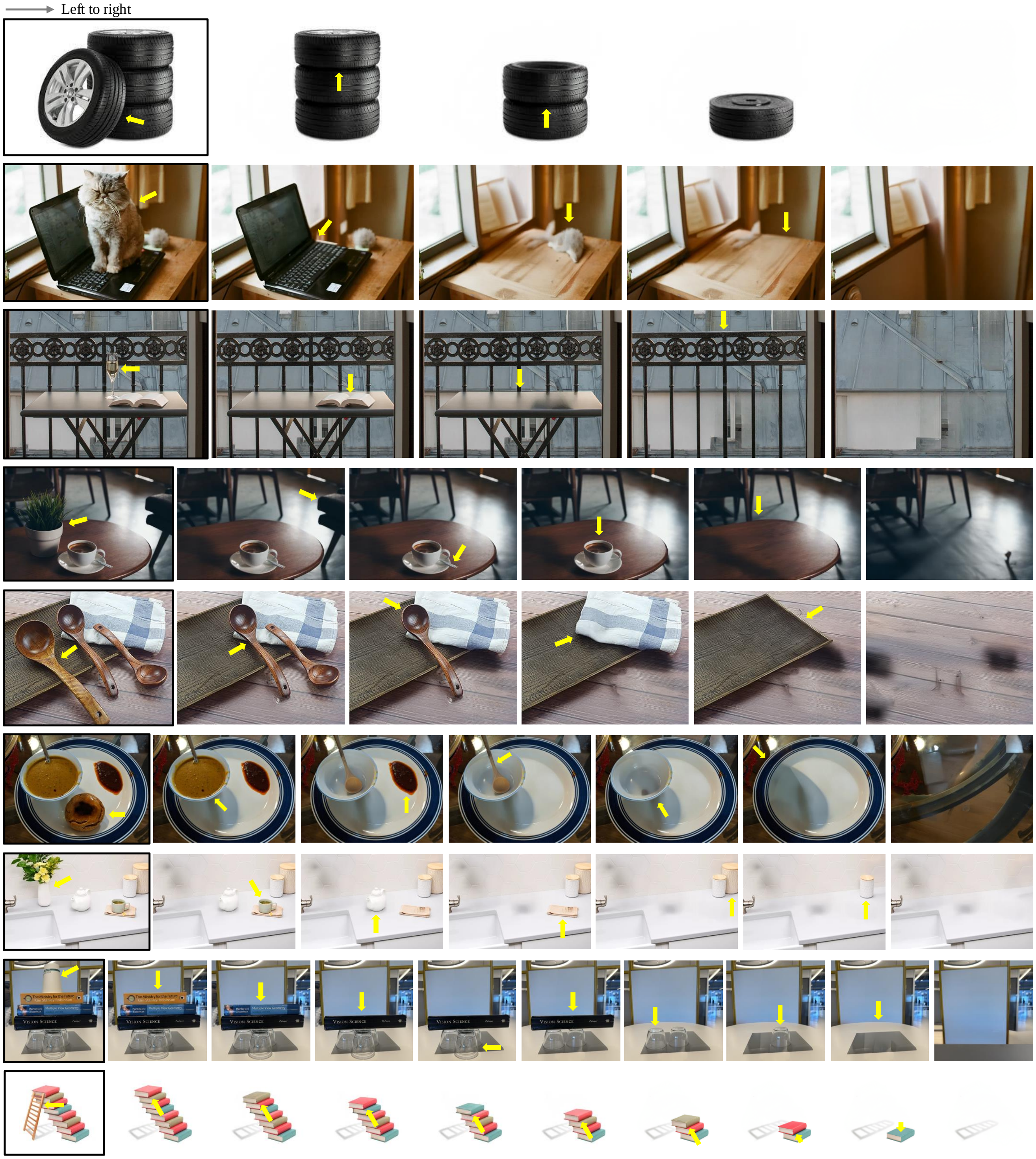


Figure 5. **Results on diverse images with increasing numbers of objects** (top to bottom). Our method produces plausible removal sequences for both simple stacked setups (first and last rows) and complex indoor scenes. For example, in the second row, the cat is removed first, followed by the laptop and table at the end. In the fifth row, the napkin is removed *after* the serving spoons and tray at the end. In the sixth row, the removal order for a dinner plate is: hard-wheat rolls (*baati*), lentil soup, sauce, spoon, and finally the plate (note that our method even removes the lentil soup). In the second-last row, note that one of three glasses is removed before the last book, which is also correct, resulting in a physically plausible sequence. For ease of visualization, we show yellow markers to highlight the object that is removed next.

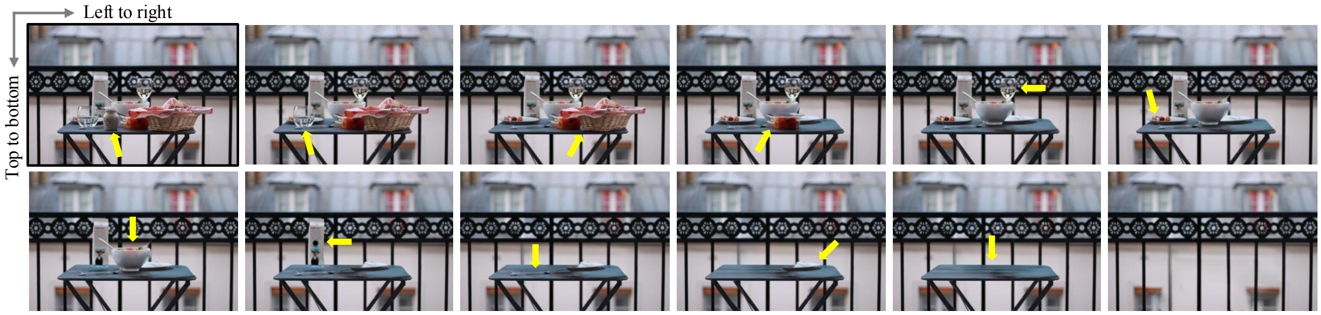


Figure 6. **Removal sequence of a breakfast table on a balcony generated by our pipeline.** Our method can accurately rank partially occluded objects, as well as a busy breakfast table setup, by sequentially removing all items. Note that new objects, such as a plate behind the basket due to occlusion, may also be introduced after an object is removed and are treated normally by our pipeline.

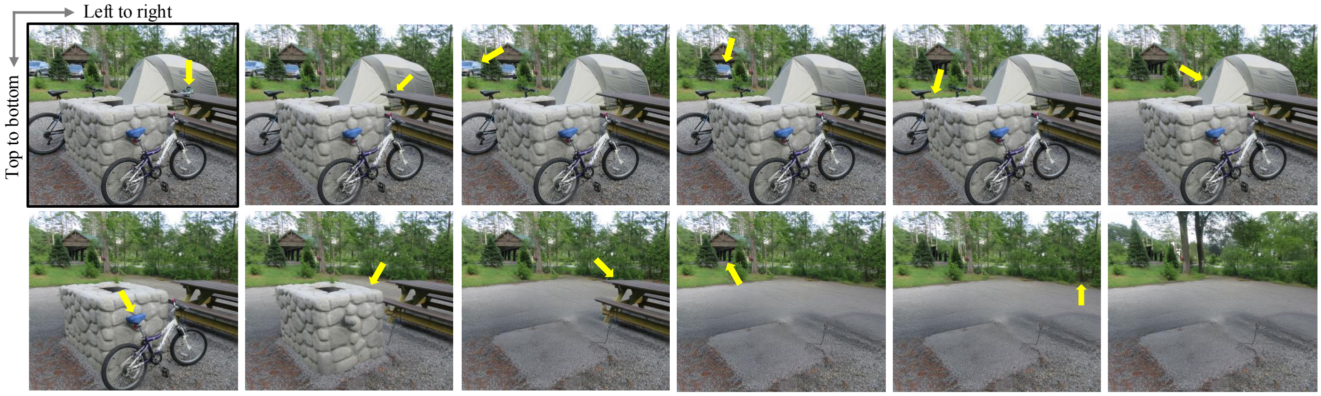


Figure 7. **Removal sequence generated by our pipeline for an outdoor scene.** Smaller objects on the table are cleared before the table itself. Additionally, the bicycles are taken away before the stone wall, which they are leaning against.

difficult HardParse dataset. Using keywords such as "messy desk," "messy room," and "stacked objects," we curated a test set of 40 challenging object pairs from 40 unique internet images, where human experts provided instance-level segmentations and non-trivial removal ordering.

Results: On the **NYU-v2**, our method correctly determined the removal order 610 of 668 pair-wise comparisons (91.32% accuracy, chance is 50%). For the 40 **HardParse** pairs, our method correctly identified 28 out of 40 pair-wise comparisons (70% accuracy, chance is 50%).

4.2. Full Scene Decomposition

While the pair-wise comparison is easy to automate, it doesn't actually model the full sequential Visual Jenga task. Therefore, we also perform a qualitative evaluation of full scene decomposition sequences. Given that scene segmentation is underdefined, e.g., segmenting a single piece of paper vs. segmenting a whole pile of papers, we require human evaluators to perform post-hoc assessments of the resulting scene decomposition sequences for overall physical plausibility and geometric coherence. To this end, we further collected another 56 unique scenes, including both our own photography and internet images, using a similar protocol to HardParse. For each of these scenes, we perform sequential object removal until only the background remains, and the human evaluator scores the whole sequences as "pass" or "fail".

Table 1. Algorithm evaluation and baseline comparisons.

Dataset Method	Pair-wise Comparison		Full Scene Decomposition
	NYU-v2 [59]	HardParse	
Top-to-Bottom	59.14%	52.25%	41.07%
Small-to-Large	90.12%	50%	42.85%
Front-to-Back	37.57%	52.5%	8.92%
Ours	**91.32%**	**65%**	**71.43%**

Results: Our method achieves 71.43% success (40/56 scenes) on full decomposition. Fig. 5 shows varying object counts, while Fig. 6, Fig. 7, and Fig. 9 show complex decompositions—including a breakfast table, an outdoor scene, and an office table, where our method successfully removes all objects (see Supp. for more results).

4.3. Comparison to simple heuristics

Given that captured visual data is often biased, we can consider how three simple object ordering heuristics perform on the Visual Jenga task. We can simply keep removing the smallest object in the scene (small-to-large heuristic), keep removing the top-most object in the image (top-to-bottom heuristic), or, with the use of single-view depth estimation [33], keep removing the closest object in the scene. As shown in Tab. 1 and in Fig. 8, these heuristics per-

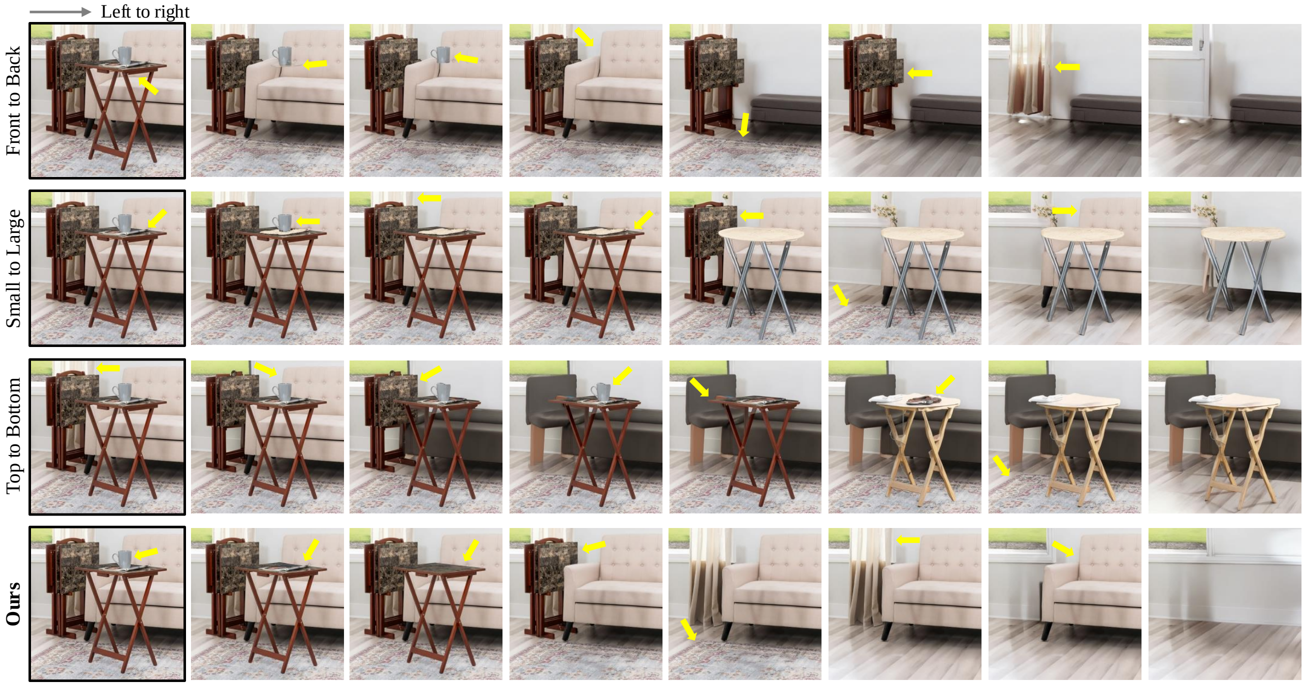


Figure 8. **Comparison against simple heuristics.** Simple heuristics, such as front-to-back ordering (top row; removes table before the cup), small-to-large ordering (second row; removes newspaper before the cup), and top-to-bottom ordering (third row; removes table before the newspaper) can sometimes work, but fail for complex real-world scenes. In contrast, our pipeline (bottom row) removes objects in a physically and semantically coherent order: first the cup, followed by the newspaper, tray table, folded tray table, curtains, floor mat, and finally the sofa.

form inconsistently across datasets—achieving high accuracy in simple scenarios (e.g., NYU-v2) but failing in more complex settings, especially in challenging datasets such as complete scene decomposition and pair-wise HardParse. While our method achieves success in more than 70% of the cases in complete scene decomposition, the best heuristic approach, small-to-large, achieves only a 43% success rate, with front-to-back performing particularly poorly at 8.9%.

4.4. Ablation studies

Diversity score. We compare different ways of qualifying semantic diversity, CLIP or DINO or both, which is used for calculating the diversity score. Tab. 2 shows that using both CLIP and DINO together gives substantially better performance especially on HardParse. Note that HardParse was never used for hyperparameters selection.

Effect of the number of inpainting samples. The larger number of inpaintings (N) helps better capture the distribution of possible scene completions and monotonically increases the performance, but with diminishing returns beyond $N = 8$ (see Supp.). We used $N = 16$ by default.

5. Discussion

In this paper, we define Visual Jenga as a purely visual task: both the input and the outputs are images. While a text-output version of our task is conceivable, one needs to be aware that it could enable shortcut solutions that don't require actual image understanding. Text outputs like "remove the book on the table" do not localize objects, and as demonstrated by Xiao et al. [62], text-based visual question-answering systems often produce plausible-sounding answers without properly grounding them in visual evidence. For Visual Jenga to meaningfully assess physical understanding, outputs must demonstrate precise spatial reasoning, which is difficult to achieve in a purely textual domain. We did experiment with running several Large Vision/Language Models on our task, followed by tools like InstructPix2pix [9] and DALL-E, to generate scene decompositions. While the LVMs often produced good textual descriptions of removal order, but we found the final visual results to be largely unusable (see Appendix G).

Table 2. Diversity score ablation: CLIP vs. DINO.

Method \ Dataset	NYU-v2	HardParse
Ours (w/o DINO)	89.52%	55%
Ours (w/o CLIP)	90.27%	57.5%
Ours (full)	**91.32%**	**65%**

While our counterfactual inpainting approach reveals how generative models capture object support dependencies, it has limitations. The current approach is slow, re-

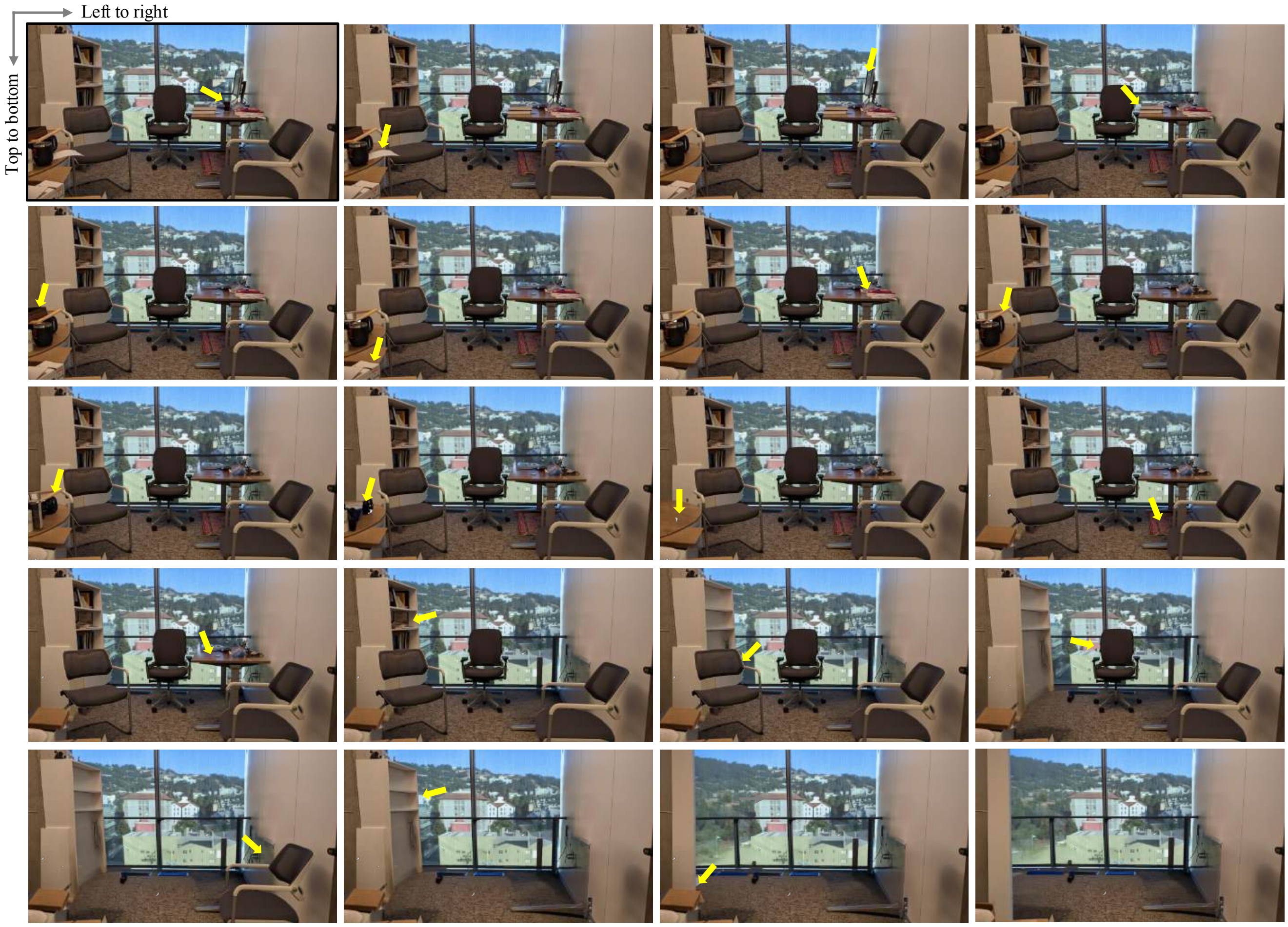


Figure 9. **Emptying an entire office room.** Our method accurately removes all objects from a complex arrangement, reliably removing items such as the carpet mat under the table and books from the bookshelf before removing the bookshelf itself, while preserving physical plausibility. Better viewed in our animated video included in Supp.

quiring multiple inpainting passes per object, and produces only a sequential removal order; unlike humans, who can envision multiple plausible removal strategies simultaneously. Our method relies on Molmo and SAM for object detection and fails when any of these models fail (Fig. 10). Rather than using a pipeline, it might be fruitful to investigate an end-to-end approach, where counterfactual reasoning is used for both, object segmentation [48] as well as scene parsing. Moreover, our approach lacks explicit physical reasoning, relying solely on statistical co-occurrences. It might be valuable to compare with methods that incorporate physical and causal reasoning — such as modeling interventions ($P(A \mid \mathrm{do}(\neg B))$) [52] — potentially by leveraging video generative models as world simulators [4, 10]. Finally, Visual Jenga also raises fundamental questions about object granularity: should an object be defined as a single sheet of paper or as an entire stack? Understanding how generative models internally represent compositional structure could offer deeper insights into object perception and reasoning.

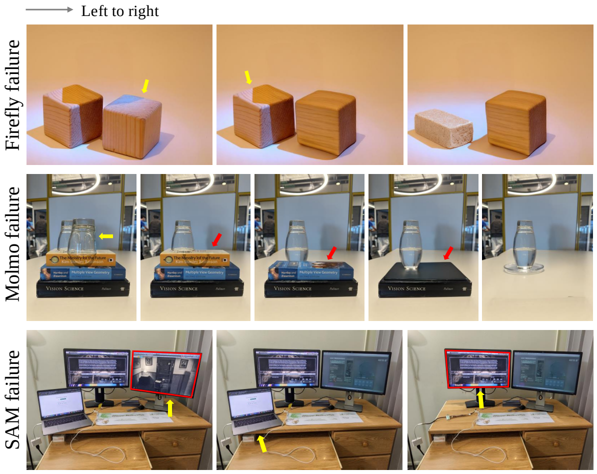


Figure 10. **Failure cases.** Our pipeline can fail for several reasons. *Top Row:* strong shadow cues near an object mislead Firefly, causing it to reinsert a new object rather than remove it. *Middle Row:* our ranking is wrong (indicated by →) because Molmo fails to identify the third glass, leaving it behind. *Bottom Row:* SAM segments only the monitor's screen (indicated by a red polygon) rather than the entire monitor, preventing its removal.

Acknowledgements: We are grateful to Yossi Gandelsman and Yutong Bai, whose prior experience with this problem and thoughtful insights have guided our work. We thank Qianqian Wang for her suggestions on our experiments, Amil Dravid and Xiao Zhang for their feedback on the manuscript, and Krishna Kumar Singh for helping us figure out object removal using Firefly. We also thank Leon Bottou for providing the initial inspiration [43] for this project, and for suggesting a connection to Reichenbach's principle [55] and probabilistic causation [27]. This research has been supported by ONR MURI grant. The work was initiated while AB was a summer visitor at UC Berkeley, supported by a grant from TRI.

References

[1] Adobe Inc. Adobe firefly - free generative AI for creatives. `https://www.adobe.com/products/firefly.html`. Accessed: 2025-3-4. 4

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[3] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros. Visual prompting via image inpainting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[4] Daniel M Bear, Kevin Feigelis, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel L K Yamins. Unifying (machine) vision via counterfactual world modeling. *arXiv preprint arXiv*, 2023. 3, 8

[5] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *International Conference on Learning Representations (ICLR)*, 2020. 2

[6] Anand Bhattad, Daniel McKee, Derek Hoiem, and David Forsyth. Stylegan knows normal, depth, albedo, and more. *Advances in Neural Information Processing Systems*, 2023. 3

[7] Anand Bhattad, James Soole, and David A Forsyth. Stylitgan: Image-based relighting via latent control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[8] Irving Biederman. On the semantics of a glance at a scene. In *Perceptual Organization*. Routledge, 1981. 1, 2

[9] T Brooks, A Holynski, and A A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 7, 2

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 8

[11] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 3

[12] Alper Canberk, Maksym Bondarenko, Ege Ozguroglu, Ruoshi Liu, and Carl Vondrick. EraseDraw: Learning to insert objects by erasing them from images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

[13] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yensung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. *arXiv [cs.CV]*, 2024. 4, 2

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009. 3

[15] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2

[16] Xiaodan Du, Nicholas Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let's find out! *arXiv preprint arXiv:2311.17137*, 2023. 3

[17] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 1

[18] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A Efros. Blobgan: Spatially disentangled scene representations. In *European conference on computer vision*. Springer, 2022. 3

[19] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 2023. 3

[20] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2024. 3

[21] Clark Glymour and Frederick Eberhardt. Hans Reichenbach. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition, 2022. 3

[22] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *In-*

ternational Conference on Machine Learning. PMLR, 2019. 2

[23] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*. PMLR, 2019. 3

[24] Ruiqi Guo and Derek Hoiem. Support surface prediction in indoor scenes. In *2013 IEEE International Conference on Computer Vision*. IEEE, 2013. 2

[25] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision(ECCV)*, 2010. 2

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. 3

[27] Christopher Hitchcock. Probabilistic Causation. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2021 edition, 2021. 3, 9

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 4

[29] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering surface layout from an image. *Int. J. Comput. Vis.*, 2007. 2

[30] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 2011. 2

[31] Wen Huang, Hongbin Liu, Minxin Guo, and Neil Zhenqiang Gong. Visual hallucinations of multi-modal large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. 3

[32] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *ICCV*, 2013. 2

[33] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 6

[34] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4

[35] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019. 2

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *International Journal of Computer Vision*. Springer, 2016. 3

[37] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016. 2

[38] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[39] Wenbin Li, Ales Leonardis, and Mario Fritz. Visual stability prediction for robotic manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2606–2613. IEEE, 2017. 1, 2

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014. 3

[41] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, (2018), 2018. 3

[42] Zhengzhe Liu, Qing Liu, Chirui Chang, Jianming Zhang, Daniil Pakhomov, Haitian Zheng, Zhe Lin, Daniel Cohen-Or, and Chi-Wing Fu. Object-level scene deocclusion. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. 2

[43] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Schölkopf, and Léon Bottou. Discovering causal signals in images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 9

[44] Tomasz Malisiewicz and Alyosha Efros. Beyond categories: The visual memex model for reasoning about object relationships. *Adv. Neural Inf. Process. Syst.*, 2009. 3

[45] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3D decomposition by rendering primitives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[46] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 2016. 2

[47] Changsuk Oh, Dongseok Shim, Taekbeom Lee, and H Jin Kim. Object remover performance evaluation methods using class-wise object removal images. *IEEE Sensors Letters*, 2024. 3

[48] Deniz Oktay, Carl Vondrick, and Antonio Torralba. Counterfactual image networks. 2018. 3, 8

[49] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features

without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. 4

[50] 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 (CVPR)*, 2024. 3

[51] Dim P Papadopoulos, Youssef Tamaazousti, Ferda Ofli, Ingmar Weber, and Antonio Torralba. How to make a pizza: Learning a compositional layer-based gan model. In *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[52] Judea Pearl. *Causality: Models, reasoning, and inference*. Cambridge University Press, Cambridge, England, 2000. 8

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 4

[54] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 4

[55] Hans Reichenbach. *The Direction of Time*. University of California Press, 1956. 3, 9

[56] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[58] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *BMVC*, 2022. 3

[59] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2, 4, 6

[60] Vaibhav Vavilala, Seemandhar Jain, Rahul Vasanth, Anand Bhattad, and David Forsyth. Blocks2world: Controlling realistic scenes with editable primitives. *arXiv preprint arXiv:2307.03847*, 2023. 2

[61] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

[62] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 7, 2

[63] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 3

[64] Michael Ying Yang, Wentong Liao, Hanno Ackermann, and Bodo Rosenhahn. On support relations and semantic scene graphs. *ISPRS J. Photogramm. Remote Sens.*, 2017. 4, 1

[65] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3

[66] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[67] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. A general protocol to probe large vision models for 3D physical understanding. In *Advances in Neural Information Processing Systems 37*, 2025. 3

[68] Chuanxia Zheng, Duy-Son Dao, Guoxian Song, Tat-Jen Cham, and Jianfei Cai. Visiting the invisible: Layer-by-layer completed scene decomposition. *International Journal of Computer Vision*, 2021. 2

[69] Liang Zhou, Kevin A Smith, Joshua B Tenenbaum, and Tobias Gerstenberg. Mental jenga: A counterfactual simulation model of causal judgments about physical support. *Journal of Experimental Psychology: General*, 2023. 3

[70] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

Visual Jenga: Discovering Object Dependencies via Counterfactual Inpainting

Supplementary Material

A. Result compilation video

We provide a result compilation video of Visual Jenga, showcasing the solutions discovered by our proposed method across various scenes in the project page.

B. Full dataset availability

All images in our evaluation datasets are provided as HTML webpages in the project page for comprehensive inspection. This includes:

1. **Full Scene Decomposition dataset:** 56 scenes collected both from our own photography and from internet searches using keywords such as "messy desk", "messy room", and "stacked objects". For each scene, we perform sequential object removal until only the background remains.
2. **Pair-wise object ordering dataset:**
    - **NYU-v2:** The NYU Depth V2 dataset contains 1449 original images. Using support relation annotations from Yang et al. [64], we extracted 485 unique images yielding 668 pair-wise comparisons with unambiguous removal ordering. Due to the limitation of the class-level (rather than instance-level) support relationship annotations from Yang et al., we carefully filtered the dataset to only include unambiguous cases. The original support label annotations from the NYU Depth V2 dataset are no longer accessible online. Despite our best efforts to contact the original authors and others who had access to the annotations, we were only able to obtain the data with difficulty. Unfortunately, the knowledge required to interpret and utilize these labels has been lost over time. Consequently, we opted to use the alternative annotations provided by Yang et al.
    - **HardParse:** Because NYU-v2 dataset has very few examples of complex object dependencies (e.g. stacks of objects, hanging/leaning objects), we produced a more difficult dataset of 40 challenging object pairs from 40 unique internet images. Using keywords such as "messy desk", "messy room", and "stacked objects", we curated this test set where human experts provided instance-level segmentations and non-trivial removal ordering.

As shown in Figure 11, we provide examples from both the NYU-v2 and HardParse pair-wise datasets. In these examples, the model is presented with an image and two segmentation masks (A and B), and must determine which object should be removed first. In both cases shown, object A is correctly identified as the one to remove first.

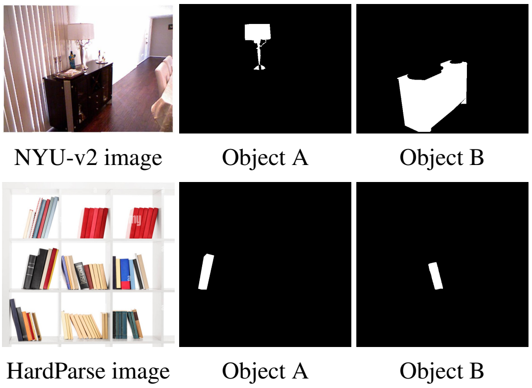
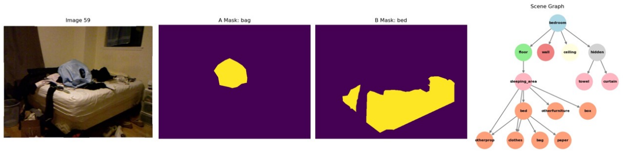


Figure 11. Examples from NYU-v2 and HardParse pair-wise sets. A model is shown an image and the segmentation masks A and B. The model determines which object should be removed first. In both examples, A is removed first.

C. Example of NYU-v2 pair-wise dataset



From left to right: (1) Original RGB image showing a bedroom scene, (2) Choice A: segmentation mask for the pillow, (3) Choice B: segmentation mask for the bed, and (4) Scene graph representation showing support relationships obtained from Yang et al. [64]. The scene graph indicates that the pillow is supported by the bed. This example demonstrates how we extract unambiguous pair-wise removal orderings from the annotated support relations in the dataset. Note that the scene graphs from Yang et al. are class-level rather than instance-level annotations, which can be ambiguous in scenes with multiple instances of the same class. We carefully filter out such ambiguous cases and only include examples where the support relationship is unambiguous. Note that the model is provided with only (1), (2), and (3), not the scene graph (4), and must make the decision between removing choice A (pillow) versus choice B (bed) first.

D. Inpainting detials

All crops are square-shaped, resized to 224 × 224 as required by CLIP and DINO, and zero-valued outside the segmentation area. Note that this measure only requires an inpainting model, not necessarily a text-to-image model. We make the best efforts to reduce textual biases in a T2I model with a generic prompt "Full HD, 4K, high quality, high resolution, photorealistic". We use the following generic, widely-used, negative prompt: “bad anatomy, bad proportions, blurry, cropped, deformed, disfigured, duplicate, error, extra limbs, gross proportions, jpeg artifacts, long neck, low quality, lowres, malformed, morbid, mutated, mutilated, out of frame, ugly, worst quality”.

E. Ablation on the similarity metrics

In addition to the quantitative results provided in Tab. 2, we also present qualitative ablation for using both CLIP and DINO scores for our ranking. In Fig. 13, we show the removal sequence when not using the CLIP scores, and in Fig. 14, we show the removal sequence when not using the DINO scores. The combination of both CLIP and DINO together gives substantially better performance, particularly on HardParse. Since HardParse was used for hyperparameter selection, this suggests that our decision choices are generalizable across scene types.

F. Ablation on the number of inpainting samples

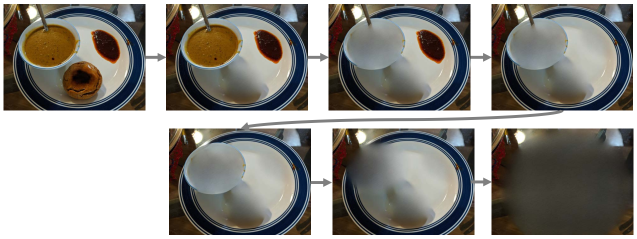
The larger number of inpaintings (N) helps better capture the distribution of possible scene completions and monotonically increases the performance, but with diminishing returns beyond $N = 8$. We used $N = 16$ by default. The performance on HardParse across different values of N is shown in the following table:

$N = 2$	$N = 4$	$N = 8$	$N = 16$
50%	50%	62.5%	65%

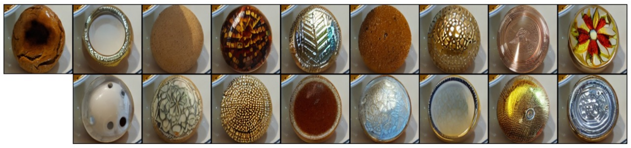
G. VLM baselines

We explore VLM-based solutions for Visual Jenga, noting that VLMs don’t directly output image sequences. As discussed in the main text, purely text-based solutions risk enabling shortcuts without true image understanding, as text outputs like "remove the book on the table" lack precise object localization and spatial reasoning [62]. We propose integrating ChatGPT 4o (November 2024) as a strong VLM baseline through three pipelines:

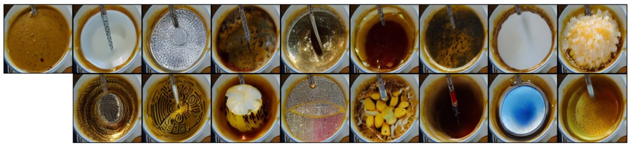
1. **ChatGPT + DALLE**: Direct image to image sequence generation as described in Sec. G.1.
2. **ChatGPT + InstructPix2Pix** [9]: Image to image editing with text prompts like "Remove <object>" as described in Sec. G.2.
3. **ChatGPT + Molmo + SAM + Adobe Firefly** [13]: Translates text to visual outputs through object localization and segmentation as described in Sec. G.3, offering a similar pipeline to ours.



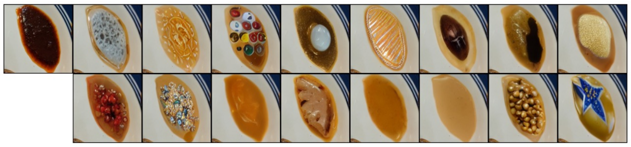
(a) Object Removal Order when emptying a food plate.



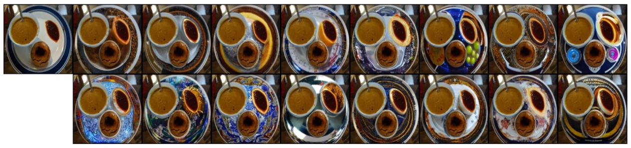
(b) Dumpling: Variance in Dumpling’s replaceability is high with many different object types and hence it is removed first.



(c) Soup in the bowl: The soup in the bowl can be replaced with many different soups, curd, milk, or other fluid types. But based on our scoring it is the second choice for removing.



(d) Red Sauce



(e) Plate has vert limited replacements possible and hence it is the last thing that is removed from the scene.

Figure 12. Visualization of object replaceability through multiple inpainting variations. The original object (a) and three different inpainting results (b-d) demonstrate the range of possible replacements while maintaining scene coherence. Higher visual diversity in replacements indicates greater replaceability.

Through extensive experimentation with different text prompts, we found that ChatGPT generally identifies correct object removal orders when interpreted by humans. However, these textual descriptions can still be ambiguous when translated into precise spatial locations. The key distinction is that our vision-based approach works directly on object segments, while the VLM methods must first translate textual outputs into spatial locations on the scene.

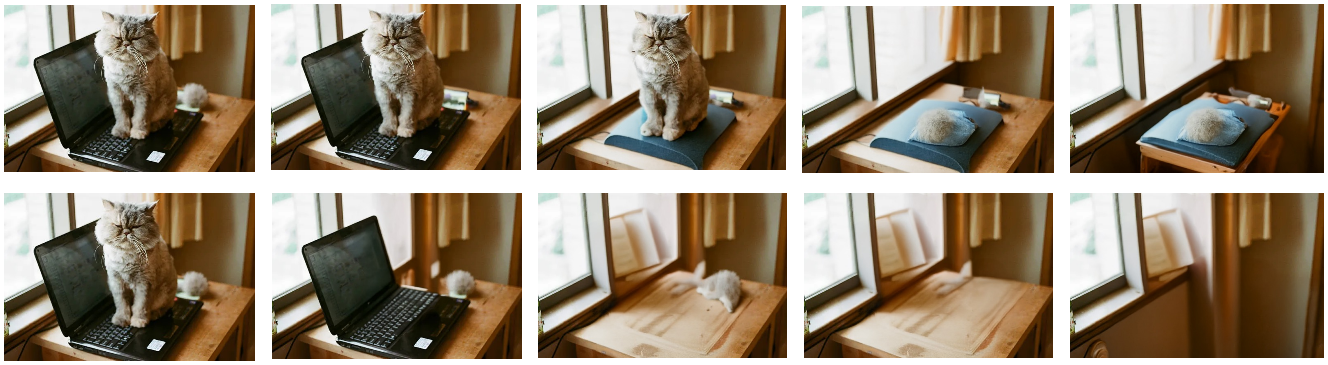


Figure 13. **Ablation: without using CLIP scores.** In the top section, we show the removal sequence without using CLIP scores. In the bottom section, we show the results when both DINO and CLIP scores are used. We observe that DINO tends to favor smaller objects. When CLIP scores are not included, the ordering can be incorrect.

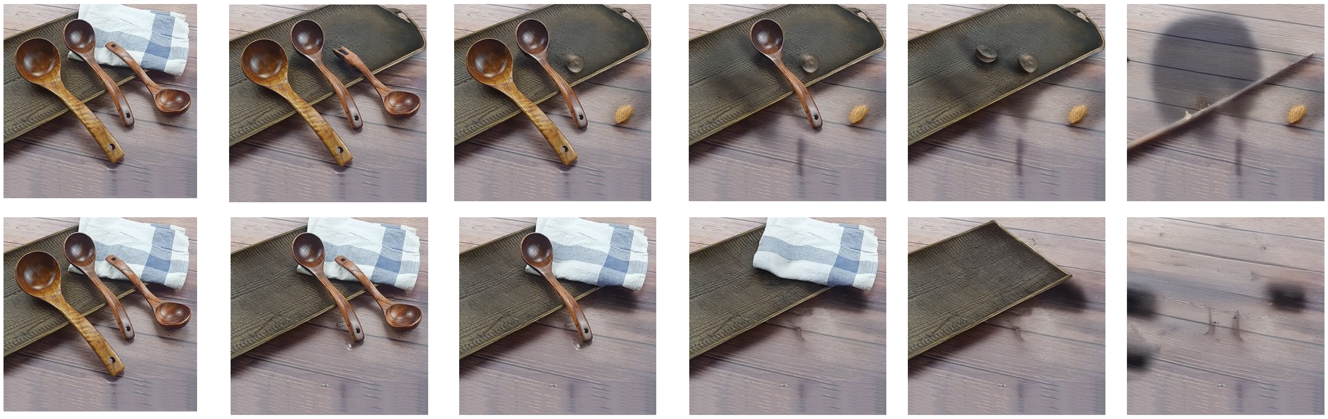


Figure 14. **Ablation: without using DINO scores.** In the top section, we show the removal sequence without using DINO scores. In the bottom section, we show the results when both DINO and CLIP scores are used. We observe that CLIP tends to overlook thin structures. When scoring between the crops, it still recognizes the spoon on top and assigns a high similarity score, which leads to the napkin being removed first.

G.1. ChatGPT + DALLE

ChatGPT Prompt

"Look at the image I gave. Find object to remove such that after removed the stability of the scene is maintained. The scene is stable if we hypothetically play the scene forward in time, the scene still maintains the same configuration. Then, use your image generation capabilities and give me the image after the object is removed."

See the results in Fig. 17.

Input image 1—Output 1:

> Based on your request, I will identify an object in the scene that can be removed while maintaining the stability of the scene. Then, I'll generate an updated image reflecting the removal of the chosen object.
>
> To proceed with the removal and generation of the updated image, I'll analyze the image for objects that are not structurally essential (like chairs or desks), aren't load-bearing, and whose absence won't disrupt the balance of the scene. Let's begin.
>
> <generated image (see Fig. 17b>
>
> Here is the updated image with the clutter removed from the floor while maintaining the stability and overall configuration of the scene.

Input image 1—Output 2:

> <generated image (see Fig. 17c)>
>
> Here is the updated image with the specified object removed while maintaining the stability and aesthetics of the scene.

Input image 1—Output 3:

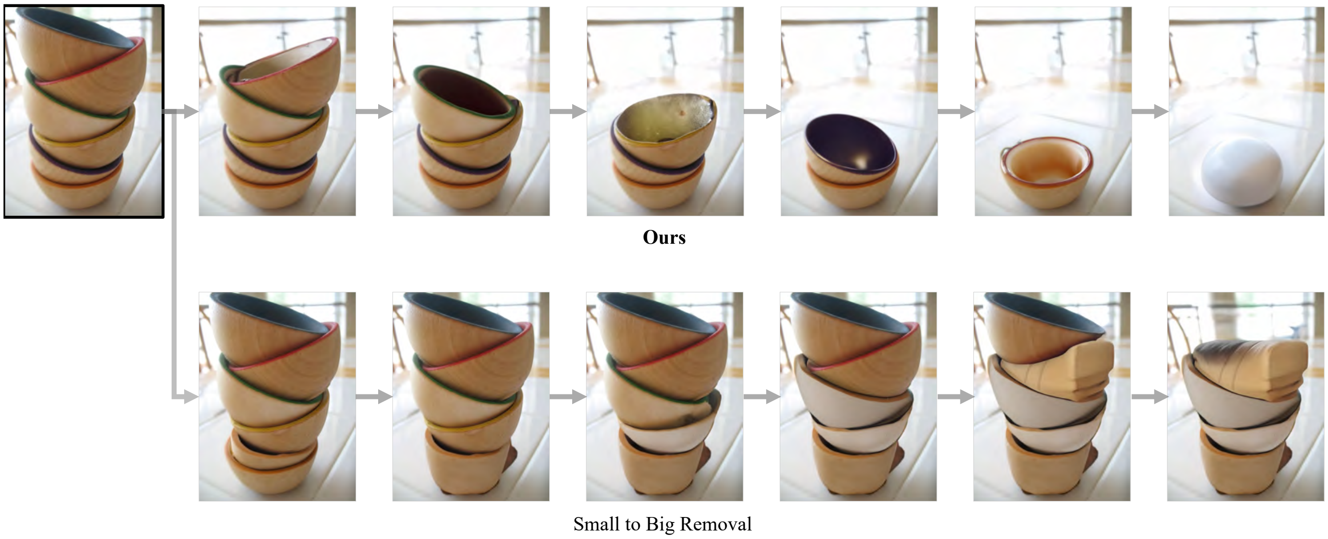


Figure 15. A failure case of a small-to-large heuristic (in the bottom). A heuristic approach may work in a few cases but fails in many other cases. Ours (in the top) fail on the last removal because strong shadow force to add objects instead of removing them.

> <generated image (see Fig. 17d)>
>
> Here is the modified image of the office after removing clutter while maintaining the scene's stability.

Input image 2—Output 1:

> Analyzing the image, it appears that the blue-and-white napkin is not critical to the structural stability of the scene and can be removed without impacting the arrangement of the objects. I will proceed to generate an image with the napkin removed.
>
> <generated image (see Fig. 17f>
>
> The image has been generated with the napkin removed, maintaining the stability of the scene. Let me know if there are further adjustments you'd like!

Input image 2—Output 2:

> Based on the image, one object that can be removed to maintain the stability of the scene is the cloth in the background. It is not directly supporting or interacting with other objects like the spoons or tray. Removing it will not destabilize the scene.
>
> Let me generate an image for you with the cloth removed.
>
> <generated image (see Fig. 17g)>
>
> Here's the updated image with the cloth removed for maintaining scene stability. Let me know if you need further adjustments!

Input image 2—Output 3:

> The image shows three wooden spoons of different sizes placed on a surface alongside a napkin. To maintain the stability of the scene, removing one of the spoons or the napkin can preserve the balance since the remaining objects would stay in place.
>
> I will remove the napkin as it is not critical for the balance of the scene. Let me generate the updated image.
>
> <generated image (see Fig. 17h)>
>
> Here is the updated image with the napkin removed while maintaining the stability of the scene.

G.2. ChatGPT + InstructPix2Pix

> **ChatGPT prompt:** "Give me the sequence order in which I should remove all objects from the scene until I am left with the background. Don't group objects. Each object should be removed one at a time and shall not disturb the stability of the other objects."
> **InstructPix2Pix prompt:** "Remove <the first object>".

See the results in Fig. 18.
Input image 1—ChatGPT output:
For the InstructPix2Pix image, see Fig. 18a.

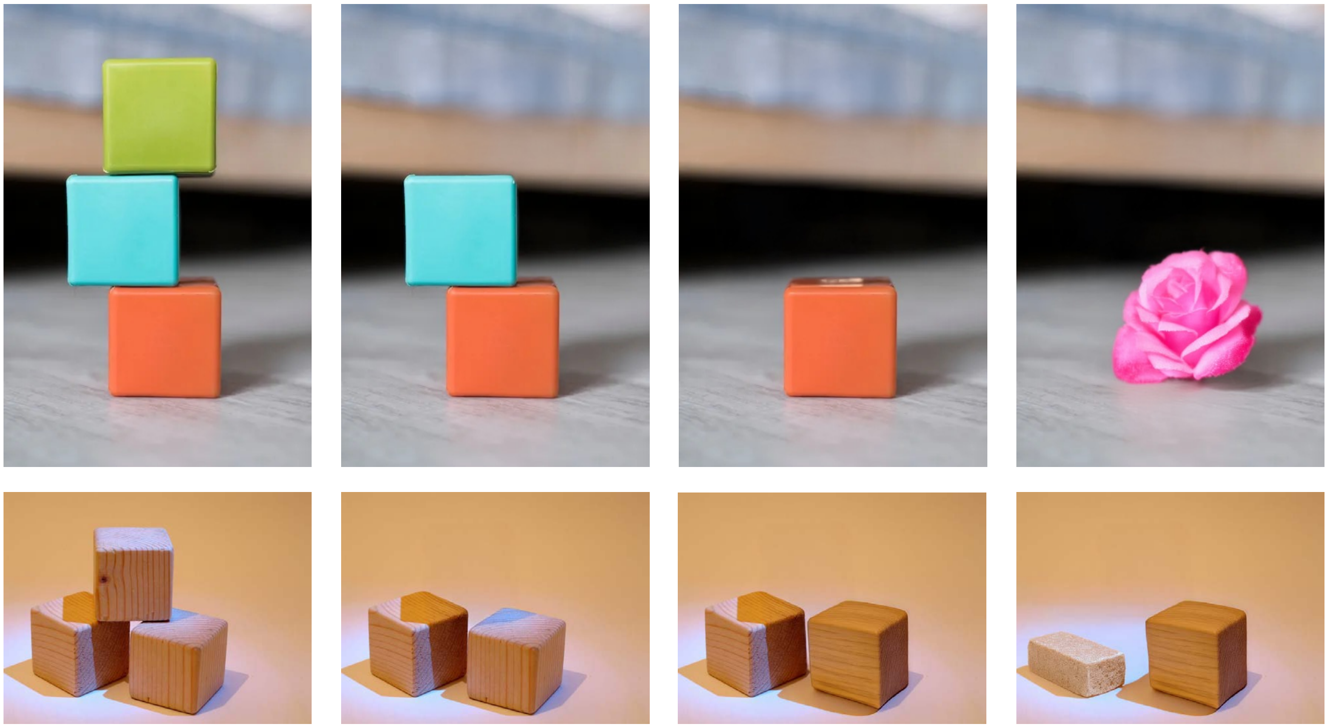
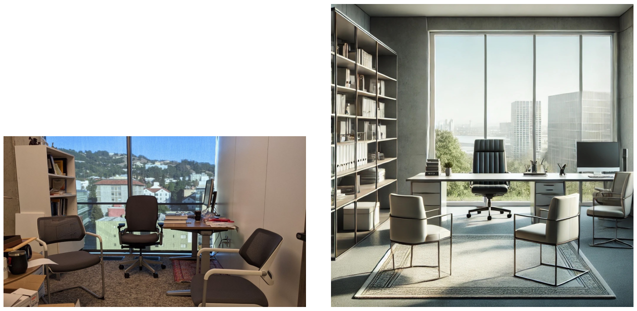


Figure 16. **Failure cases due to (Shadow Residuals).** Even when the sequence is correct, strong shadow cues can lead to incomplete removal. Instead of removing the object, Adobe Firefly responds to these cues and adds a new object instead.

> To remove objects one at a time without disturbing the stability of other objects, here's the suggested sequence of removal:
>
> 1. **Papers on the desk**: Start with the loose papers on the desk to reduce clutter.
> 2. **Mugs on the desk and side table**: Remove the mugs next, as they are small and easy to move.
> 3. **Glasses on the desk**: Take the glasses off the desk.
> 4. **Books and files on the desk**: Remove these one by one carefully.
> 5. **Small items (e.g., pens or small electronics)** on the desk.
> 6. **Loose items on the bookshelf**: Clear the items on the shelves, starting from the top shelf to the bottom shelf, to maintain balance.
> 7. **Chair near the desk (right-hand side)**: Move the smaller chair beside the desk.
> 8. **Chair near the side table (left-hand side)**: Remove the other smaller chair.
> 9. **Central chair**: Move the office chair in the middle.
> 10. **Rug under the desk**: Carefully lift and remove the rug.
> 11. **Side table**: Remove the side table after clearing all items on it.
> 12. **Desk**: Remove the desk after ensuring all items on and around it are gone.
> 13. **Bookshelf**: Carefully move the bookshelf after everything else is cleared.
>
> By following this order, you'll remove objects without compromising the stability of others and leave only the background view.

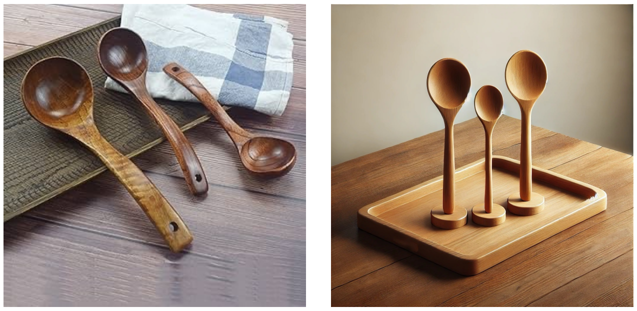
Input image 2—ChatGPT output:
For the InstructPix2Pix image, see Fig. 18b.

> To remove all objects from the scene without disturbing the stability of the other objects, follow this sequence:
>
> 1. **Remove the paper cup** (topmost object). This object is light and does not support any other objects.
> 2. **Remove the top glass** from the stack of glasses. Carefully lift the glass without disturbing the ones below it.
> 3. **Remove the second glass** from the stack.
> 4. **Remove the third glass** from the stack.
> 5. **Remove the fourth glass** (bottom-most glass in the stack).

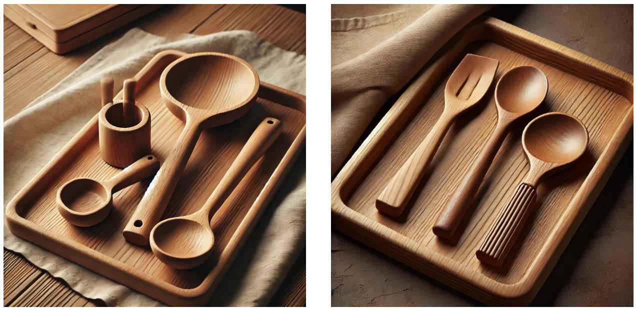


(a) **Input image 1** (b) **Output 1**

(c) **Output 2** (d) **Output 3**



(e) **Input image 2** (f) **Output 1**



(g) **Output 2** (h) **Output 3**

Figure 17. **ChatGPT + DALLE baseline.** Comparison of the original images and updated scenes with the first object removed. DALLE doesn't do a good job at preserving the original image content.

6. **Remove the book "The Ministry for the Future"** (yellow cover).
7. **Remove the book "Multiple View Geometry"** (blue cover).
8. **Remove the book "Vision Science"** (black cover).

This sequence ensures stability as you work your way from the topmost object to the base.

Input image 3—ChatGPT output:
For the InstructPix2Pix image, see Fig. 18c.

To remove objects one at a time while ensuring stability, follow this sequence:

1. **Smallest spoon on the right** - Start with the smallest object, as it does not provide structural support to other objects.
2. **Medium-sized spoon in the middle** - Next, remove the medium-sized spoon, as it is similarly isolated and won't destabilize anything.
3. **Largest spoon on the left** - Then remove the largest spoon, as it is freestanding and can be removed without affecting the others.
4. **Napkin on the top right** - Finally, remove the napkin, as it is flat and won't interfere with the background.
5. **Tray on the bottom left** - End with the tray, as it forms part of the background base and is the largest object remaining.

After following this order, you will be left with only the background.

Input image 4—ChatGPT output:
For the InstructPix2Pix image, see Fig. 18d.

Here is the sequence of removal to ensure the stability of objects and to avoid disturbing others:

1. **Knife** (on the cake with sprinkles) - It's small and easy to remove without affecting stability.
2. **Can of whipped cream** (right side) - Freestanding and easy to pick up.
3. **Pink drink in the glass** (right side) - Remove carefully to avoid any spills.
4. **Fruit plate** (back right, with kiwi and berries) - This is stable and can be picked up without disturbing others.
5. **Bowl with crumble** (back middle, near the pie) - A lightweight item that is independent.
6. **Bowl of chocolates** (front right) - Remove carefully as it doesn't seem to be stacked or linked to others.
7. **Pie** (front right in a red dish) - Freestanding and can be removed next without disturbing others.

8. **Cheesecake** (middle back, with red topping) - Positioned safely on its plate, making it stable to remove next.
9. **Cake with sprinkles** (front left) - Finally remove this large, stable item.

Now you should be left with just the background.

G.3. ChatGPT + Molmo + SAM + Adobe Firefly

ChatGPT Prompt

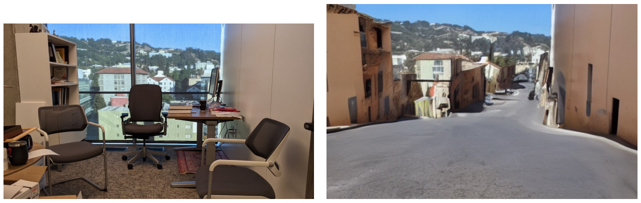
> "Give me the sequence order in which I should remove all objects from the scene until I am left with the background. Don't group objects. Each object should be removed one at a time and shall not disturb the stability of the other objects."

This pipeline is described as follows: First, obtain the removal order from ChatGPT using the prompt from Sec. G.2 above. Second, translate the textual removal order into image segmentation using Molmo to point given the object's text description, and then SAM to segment from a point. Third, remove each object in order using Adobe Firefly given the object's segmentation. These steps are important to turn textual output from ChatGPT into a visual output expected by Visual Jenga.

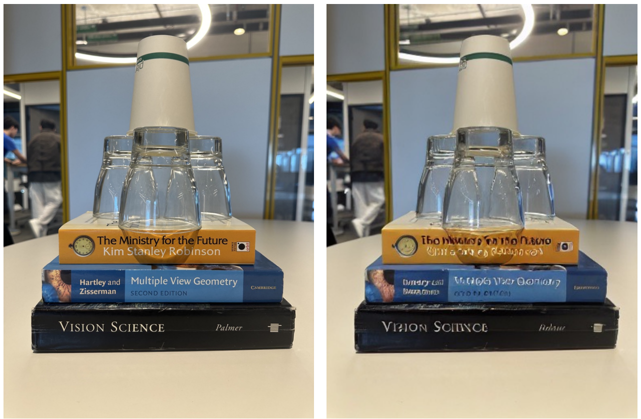
This solution can solve all simple cases. However, failure cases in Fig. 20 suggest that: First, the textual description of the object and its location can be ambiguous which leads to pointing errors from Molmo. This kind of error is more likely to happen in a scene with multiple objects of the same kind as also observed in Fig. 19 (b). Second, ChatGPT does make mistakes. Finally, we show side by side comparison with our proposed method showing different modes of failures between the two methods in Fig. 19: our proposed method may suffer from transparent objects, the ChatGPT method may suffer from ambiguous object descriptions.

Input image 1—ChatGPT output:
For the image sequence, see Fig. 19b (top).

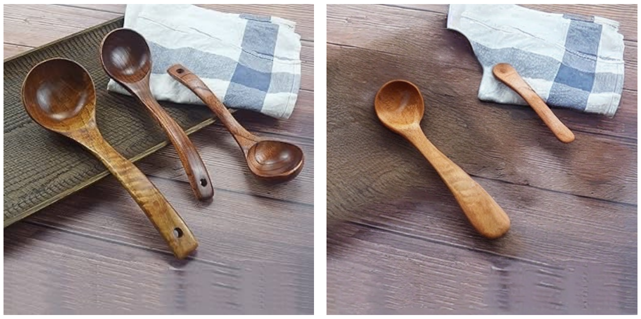
> To remove all objects from the scene without disturbing stability, follow this order:
>
> 1. Remove the **top paper cup** gently.
> 2. Remove the **book "The Ministry for the Future"** (orange book).
> 3. Remove the **book "Multiple View Geometry"** (blue book).
> 4. Remove the **book "Vision Science"** (black book).
> 5. Sequentially remove each **glass cup** one at a time from the stack (there appear to be four glass cups, so remove them one by one).


Input image **InstructPix2Pix output**

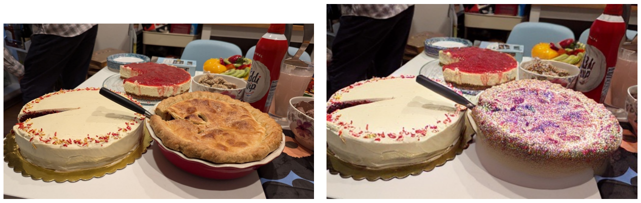
(a) **ChatGPT output.** To remove objects one at a time without disturbing the stability of other objects, here's the suggested sequence of removal: 1. Papers on the desk ...


Input image **InstructPix2Pix output**

(b) **ChatGPT output.** To remove all objects from the scene without disturbing the stability of the other objects, follow this sequence: 1. Remove the paper cup (topmost object)...

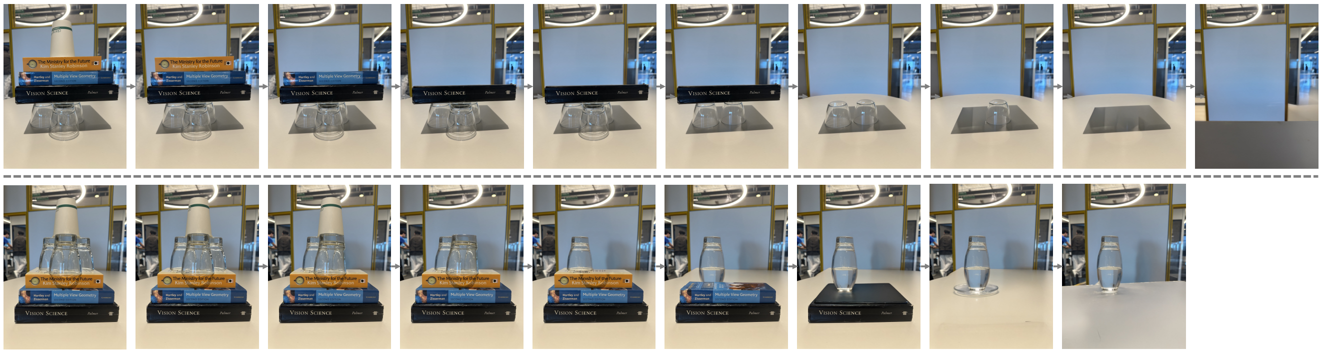

Input image **InstructPix2Pix output**

(c) **ChatGPT output.** To remove objects one at a time while ensuring stability, follow this sequence: 1. Smallest spoon on the right ...

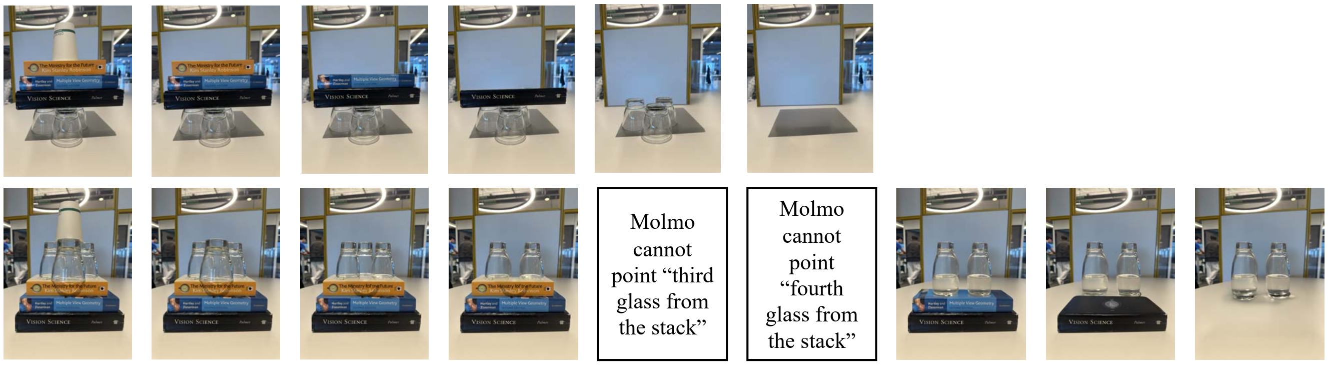

Input image **InstructPix2Pix output**

(d) **ChatGPT output.** Here is the sequence of removal to ensure the stability of objects and to avoid disturbing others: 1. Knife (on the cake with sprinkles) ...

Figure 18. **ChatGPT + InstructPix2Pix.** Comparison of the original images and updated scenes with the first object removed. InstructPix2Pix cannot follow the prompt to remove an object in the image well.



(a) **Solutions from our proposed counterfactual inpainting pipeline.**


Molmo cannot point "third glass from the stack"
Molmo cannot point "fourth glass from the stack"

(b) **Solutions from using VLM (ChatGPT 4o) with a similar pipeline (Molmo + SAM + Firefly).** For the top row, at the last step, ChatGPT suggests "Sequentially remove each glass cup one at a time from the stack (there appear to be four glass cups, so remove them one by one)", which Molmo points to all of the glass cups effectively removing them all at once. For the second row, after removing the paper cup, ChatGPT suggests: "Remove the top glass from the stack of glasses", "Remove the second glass from the stack", "Remove the third glass from the stack", and "Remove the fourth glass (bottom-most glass in the stack)". Such description can be ambiguous, and Molmo failed to locate the "third" and the "fourth" glasses as shown above.

Figure 19. **Comparing our counterfactual inpainting with a VLM-based method (ChatGPT 4o) sharing a similar pipeline.** demonstrates different failure modes from the two approaches. The ChatGPT solution has a bottleneck in coming up with a clear text description for locating an object which is exacerbated where there are many similar objects, e.g. glasses, in the scene.

> This sequence ensures the stability of the objects as you remove them.

Input image 2—ChatGPT output:
For the image sequence, see Fig. 19b (bottom).

> To remove all objects from the scene without disturbing the stability of the other objects, follow this sequence:
>
> 1. Remove the paper cup (topmost object).
> 2. Remove the top glass from the stack of glasses.
> 3. Remove the second glass from the stack.
> 4. Remove the third glass from the stack.
> 5. Remove the fourth glass (bottom-most glass in the stack).
> 6. Remove the book "The Ministry for the Future" (yellow cover).
> 7. Remove the book "Multiple View Geometry" (blue cover).
> 8. Remove the book "Vision Science" (black cover).
>
> By following this order, you'll remove objects without compromising the stability of others and leave only the background view.

Input image 3—ChatGPT output:
For the image sequence, see Fig. 20 (top).

> In the image, there are four stacked tires and a wheel. To remove all the objects one at a time without disturbing the stability of the remaining objects, follow this sequence:
>
> 1. **Top tire** - Remove the tire at the very top of the stack first.
> 2. **Second tire** - Remove the tire directly beneath the top tire.
> 3. **Third tire** - Remove the next tire in the stack.
> 4. **Bottom tire** - Remove the last tire in the stack.

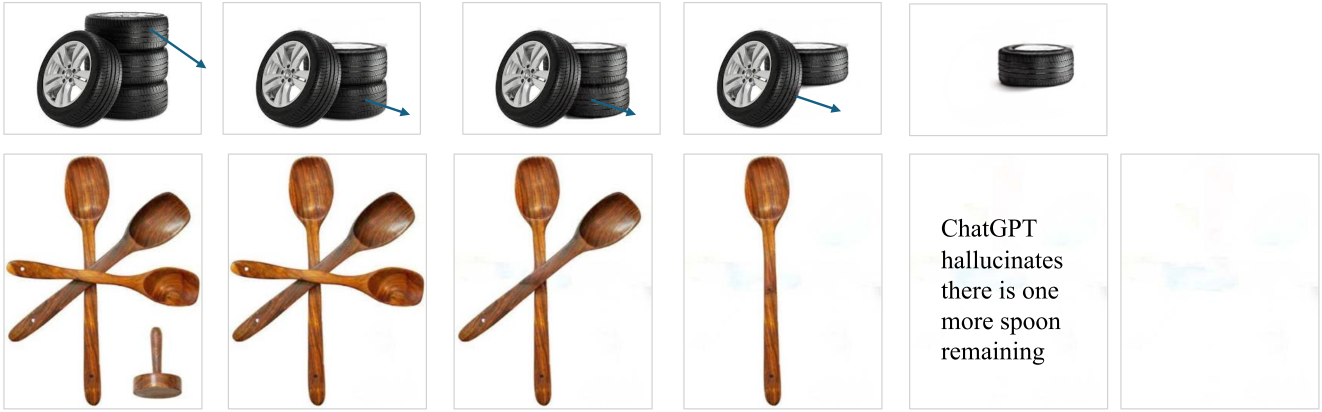


Figure 20. **Failure Cases from ChatGPT + Molmo + SAM + Adobe Firefly.** In the top row, ChatGPT recommends removing the leaning tire last. Additionally, based on its description, Molmo points to the bottom tire twice, likely due to the word “beneath." In the bottom row, ChatGPT mistakenly adds an extra spoon, while Molmo indicates the top spoon in the stack. Below are the output suggestions from ChatGPT for the top row.“In the image, there are four stacked tires and a wheel. To remove all the objects one at a time without disturbing the stability of the remaining objects, follow this sequence: 1 **Top tire**—Remove the tire at the very top of the stack first. 2. **Second tire**—Remove the tire directly beneath the top tire. 3. **Third tire**—Remove the next tire in the stack. 4. **Bottom tire**—Remove the last tire in the stack. 5. **Wheel**— Finally, remove the wheel, as it is resting separately and won’t affect the stability of any other objects. This sequence ensures that the stack’s stability is maintained during the removal process.” Below are the output suggestions from ChatGPT for the bottom row. “To ensure stability while removing the objects one at a time: 1. Remove the small cylindrical object (bottom right) as it is not supporting any other object. 2. Remove the wooden spoon leaning at the bottom right diagonally since it overlaps with the others and is not directly supporting them. 3. Remove the wooden spoon at the bottom left as it supports the least weight in the arrangement. 4. Remove the spoon on the top left that overlaps minimally with the one left behind. 5. Finally, remove the last spoon resting horizontally across the center. This sequence ensures stability and minimizes disturbance to the other objects.”

> 5. **Wheel** - Finally, remove the wheel, as it is resting separately and won’t affect the stability of any other objects.
>
> This sequence ensures that the stack’s stability is maintained during the removal process.

Input image 4—ChatGPT output:
For the image sequence, see Fig. 20 (bottom).

> To ensure stability while removing the objects one at a time:
>
> 1. Remove the **small cylindrical object** (bottom right) as it is not supporting any other object.
> 2. Remove the **wooden spoon leaning at the bottom right diagonally** since it overlaps with the others and is not directly supporting them.
> 3. Remove the **wooden spoon at the bottom left** as it supports the least weight in the arrangement.
> 4. Remove the **spoon on the top left** that overlaps minimally with the one left behind.
> 5. Finally, remove the **last spoon resting horizontally across the center**.
>
> This sequence ensures stability and minimizes disturbance to the other objects.

H. Error accumulation from multi-step solutions

Both our proposed solution and the ChatGPT solution in Sec. G.3 involve multiple steps. Each introduces its own kind of error which accumulates. At the segmentation stage, given just a point, segmentation is very much an underdetermined task, and SAM, the state-of-the-art segmentation method, is likely to make mistakes. At the object removal stage, some hard cases involve strong shadows or reflections which are not considered part of the segmentation in a traditional sense but are quite important for correct removal as a strong cue gives away the presence of the object which makes the object removal very hard or impossible. For these reasons, an end-to-end vision-based solution is highly desirable and presents a promising direction for future work.