

Supplement Material for Visua Jenga: Discovering Object Dependencies via Counterfactual Inpainting

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 Supplementary.

B Full dataset availability

All images in our evaluation datasets are provided as HTML webpages in the supplement folder.

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. [67], 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.
- **COCO:** We collected 200 images randomly from the COCO dataset (COCO 2017 train split) and used the ground truth instance segmentation that came with the dataset. Our collection methodology was as follows: (1) randomly select an image, (2) retrieve all instance segmentations in the image, (3) keep only segmentations that are not too small (larger than 1% of the total image area), (4) create all possible pairs of segmentations that are spatially next to each other (within 1 pixel radius), and (5) manually review the pairs, keeping only those with clear/unambiguous support relationships and good segmentation quality that covers most of the object area. We repeated this process for 200 images in random order to reduce selection bias. Note that a single image may contain multiple support relationship pairs; however, we tried to avoid reusing the same image multiple times in the dataset, unless the pairs demonstrated different kinds of support relationships that are visually distinct.
- **ClutteredParse:** 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 ClutteredParse 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.

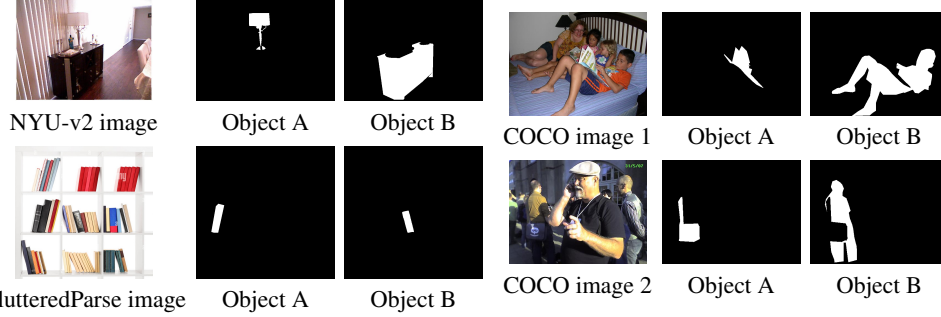
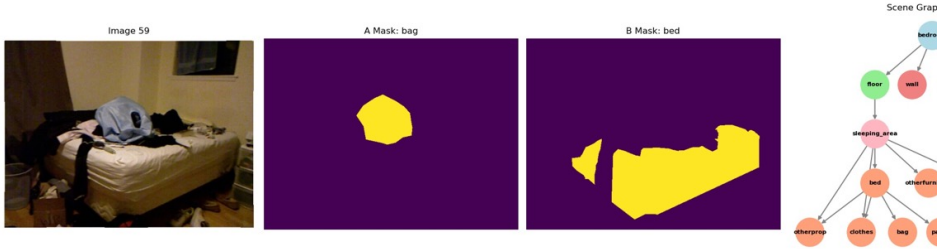


Figure 11: Examples from NYU-v2, ClutteredParse, and COCO pair-wise sets. Left: NYU-v2 and ClutteredParse examples. Right: Two examples from COCO. For each, a model is shown an image and the segmentation masks A and B, and must determine which object should be removed first.

C Example of NYU-v2 pair-wise dataset



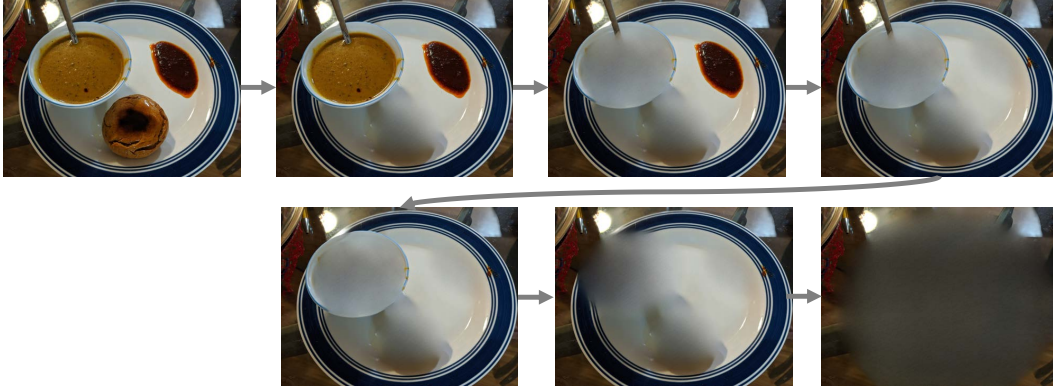
From left to right: (1) Original RGB image showing a bedroom scene, (2) Choice A: segmentation mask for the pillow, (3) Choice B: segmentation mask for the bed, and (4) Scene graph representation showing support relationships obtained from Yang et al. [67]. The scene graph indicates that the pillow is 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 details

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 Table 3, 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 ClutteredParse. Since ClutteredParse was used for hyperparameter selection, this suggests that our decision choices are generalizable across scene types.



(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.

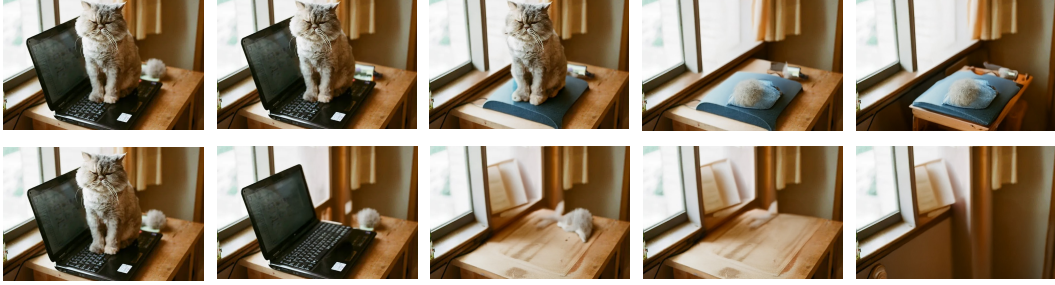


Figure 13: **Ablation: without using CLIP scores.** In the top section, we show the removal sequence without using CLIP scores. In the bottom section, we show the results when both DINO and CLIP scores are used. We observe that DINO tends to favor smaller objects. When CLIP scores are not included, the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

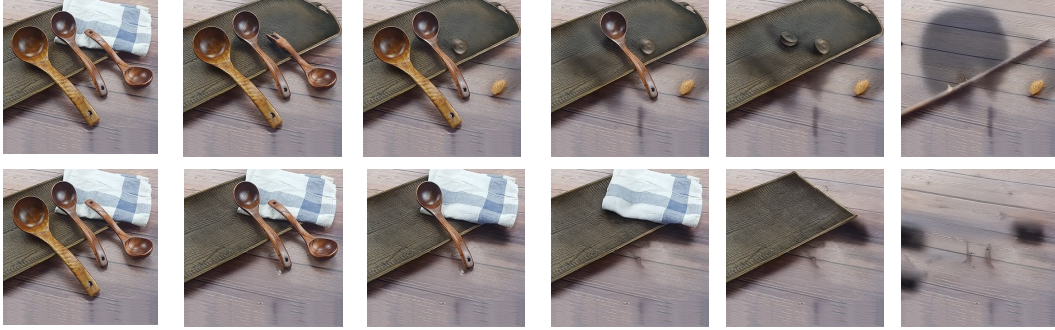


Figure 14: **Ablation: without using DINO scores.** In the top section, we show the removal sequence without using DINO scores. In the bottom section, we show the results when both DINO and CLIP scores are used. We observe that CLIP tends to overlook thin structures. When scoring between the crops, it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

872 F Ablation on the number of inpainting samples

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

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

878 G VLM baselines

879 We explore VLM-based solutions for Visual Jenga, noting that VLMs don't directly output image
 880 sequences. As discussed in the main text, purely text-based solutions risk enabling shortcuts without
 881 true image understanding, as text outputs like "remove the book on the table" lack precise object
 882 localization and spatial reasoning [64]. We propose integrating ChatGPT 4o (November 2024) as a
 883 strong VLM baseline through three pipelines:

- 884 1. **ChatGPT + DALLE:** Direct image to image sequence generation as described in Sec. ??.
- 885 2. **ChatGPT + InstructPix2Pix [10]:** Image to image editing with text prompts like "Remove
 886 <object>" as described in Sec. G.2.

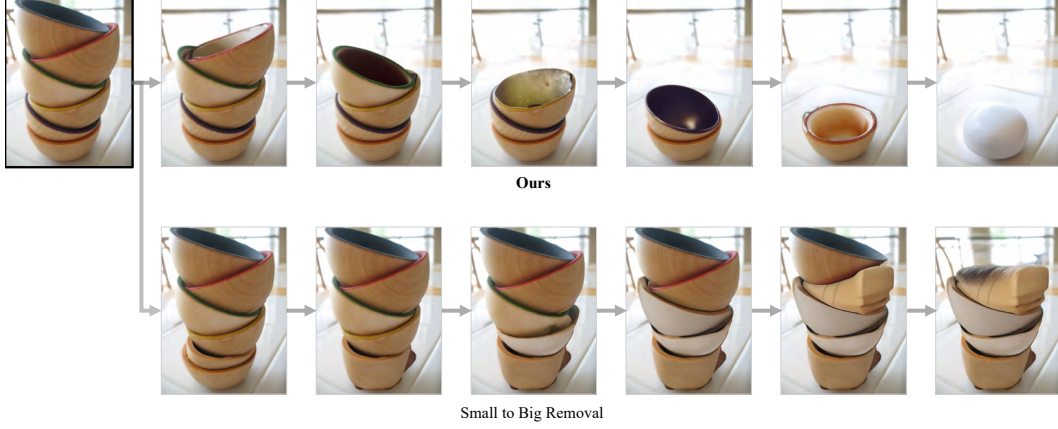


Figure 15: **A failure case of a small-to-large heuristic (in the bottom).** A heuristic approach may work in a few cases but fails in many other cases. Ours (in the top) fail on the last removal because strong shadow force to add objects instead of removing them.

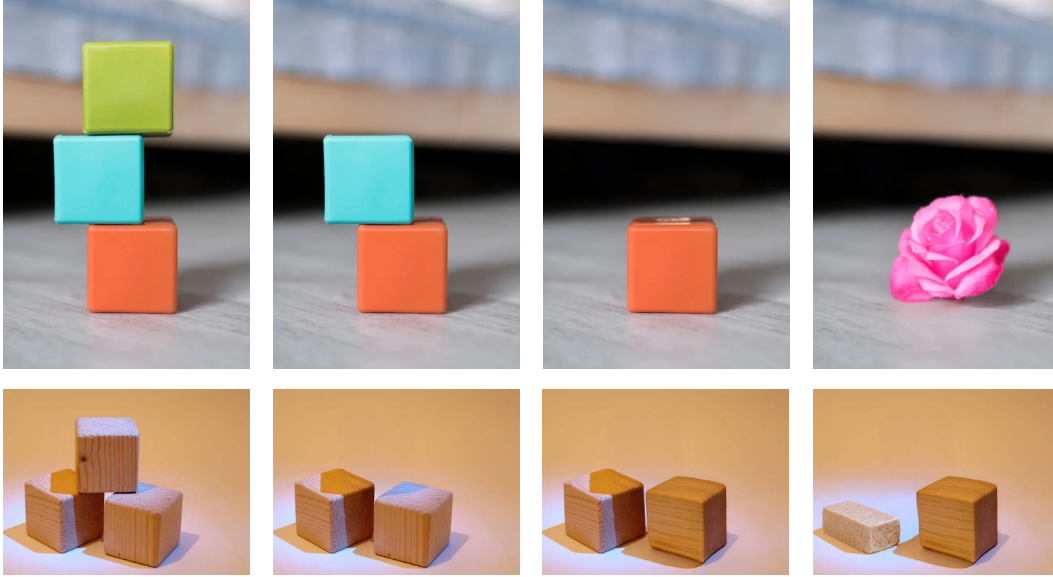


Figure 16: **Failure cases due to (Shadow Residuals).** Even when the sequence is correct, strong shadow cues can lead to incomplete removal. Instead of removing the object, Adobe Firefly responds to these cues and adds a new object instead.

887 3. **ChatGPT + Molmo + SAM + Adobe Firefly** [14]: Translates text to visual outputs through
 888 object localization and segmentation as described in Sec. G.3, offering a similar pipeline to
 889 ours.

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

895 G.1 ChatGPT 4o with image generation

896 ChatGPT 4o’s image generation capabilities [1], introduced by OpenAI on March 25, 2025, represent
 897 a significant improvement over previous models such as DALL-E. However, as we demonstrate here,



Figure 17: **ChatGPT + Image Generation [1]**. Comparison of the original images and the generated scenes after attempting to remove the first object. The ChatGPT 4o image generation model does not reliably preserve the original image content, making it difficult to determine which objects the model intended to remove.

its editing abilities on the Visual Jenga tasks still struggle to preserve the original scene content, making it difficult to determine which objects the model intended to remove.

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.

See the results in Fig. 17.

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.

See the results in Fig. 18.

Input image 1—ChatGPT output:

For the InstructPix2Pix image, see Fig. 18a.

To remove objects one at a time without disturbing the stability of other objects, here’s the suggested sequence of removal:

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

927 **Input image 2—ChatGPT output:**

928 For the InstructPix2Pix image, see Fig. 18b.

929 To remove all objects from the scene without disturbing the stability of the other
930 objects, follow this sequence:

- 931 1. **Remove the paper cup** (topmost object). This object is light and does not
932 support any other objects.
933 2. **Remove the top glass** from the stack of glasses. Carefully lift the glass
934 without disturbing the ones below it.
935 3. **Remove the second glass** from the stack.
936 4. **Remove the third glass** from the stack.
937 5. **Remove the fourth glass** (bottom-most glass in the stack).
938 6. **Remove the book “The Ministry for the Future”** (yellow cover).
939 7. **Remove the book “Multiple View Geometry”** (blue cover).
940 8. **Remove the book “Vision Science”** (black cover).

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

943 **Input image 3—ChatGPT output:**

944 For the InstructPix2Pix image, see Fig. 18c.

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

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

956 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.

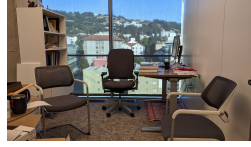
Input image 1—ChatGPT output:

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).

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

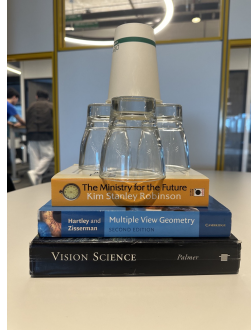


Input image

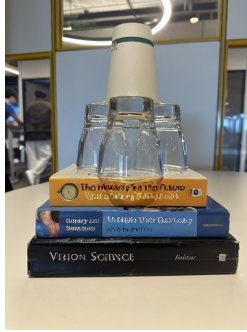


InstructPix2Pix output

(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 ...



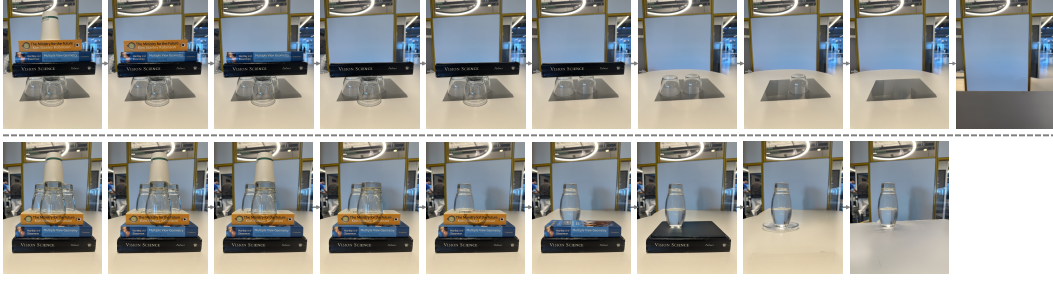
Input image



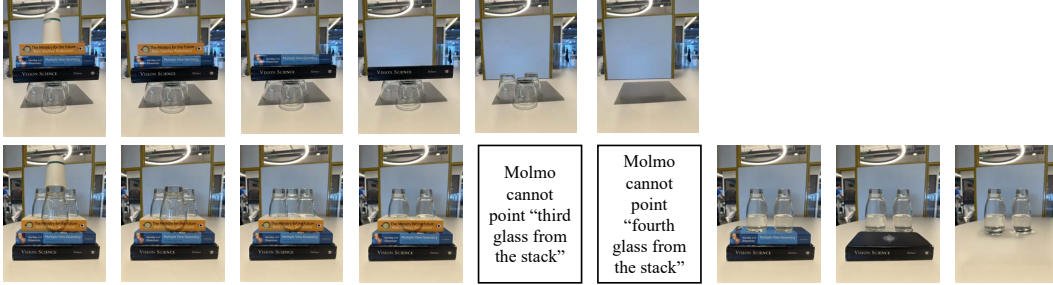
InstructPix2Pix output

(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.



(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.

1001 **Input image 2—ChatGPT output:**

1002 For the image sequence, see Fig. 19b (bottom).

1003 To remove all objects from the scene without disturbing the stability of the other
1004 objects, follow this sequence:

- 1005 1. Remove the paper cup (topmost object).
- 1006 2. Remove the top glass from the stack of glasses.
- 1007 3. Remove the second glass from the stack.
- 1008 4. Remove the third glass from the stack.
- 1009 5. Remove the fourth glass (bottom-most glass in the stack).
- 1010 6. Remove the book “The Ministry for the Future” (yellow cover).
- 1011 7. Remove the book “Multiple View Geometry” (blue cover).
- 1012 8. Remove the book “Vision Science” (black cover).

1013 By following this order, you’ll remove objects without compromising the stability
1014 of others and leave only the background view.

1015 **Input image 3—ChatGPT output:**

1016 For the image sequence, see Fig. 20 (top).

1017 In the image, there are four stacked tires and a wheel. To remove all the objects
1018 one at a time without disturbing the stability of the remaining objects, follow this
1019 sequence:



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.”

- 1020 1. **Top tire** - Remove the tire at the very top of the stack first.
- 1021 2. **Second tire** - Remove the tire directly beneath the top tire.
- 1022 3. **Third tire** - Remove the next tire in the stack.
- 1023 4. **Bottom tire** - Remove the last tire in the stack.
- 1024 5. **Wheel** - Finally, remove the wheel, as it is resting separately and won’t affect
- 1025 the stability of any other objects.

1026 This sequence ensures that the stack’s stability is maintained during the removal
1027 process.

1028 **Input image 4—ChatGPT output:**

1029 For the image sequence, see Fig. 20 (bottom).

1030 To ensure stability while removing the objects one at a time:

- 1031 1. Remove the **small cylindrical object** (bottom right) as it is not supporting
- 1032 any other object.
- 1033 2. Remove the **wooden spoon leaning at the bottom right diagonally** since it
- 1034 overlaps with the others and is not directly supporting them.
- 1035 3. Remove the **wooden spoon at the bottom left** as it supports the least weight
- 1036 in the arrangement.
- 1037 4. Remove the **spoon on the top left** that overlaps minimally with the one left
- 1038 behind.
- 1039 5. Finally, remove the **last spoon resting horizontally across the center**.

1040 This sequence ensures stability and minimizes disturbance to the other objects.

1041 **H Error accumulation from multi-step solutions**

1042 Both our proposed solution and the ChatGPT solution in Sec. G.3 involve multiple steps. Each
1043 introduces its own kind of error which accumulates. At the segmentation stage, given just a point,
1044 segmentation is very much an underdetermined task, and SAM, the state-of-the-art segmentation
1045 method, is likely to make mistakes. At the object removal stage, some hard cases involve strong
1046 shadows or reflections which are not considered part of the segmentation in a traditional sense but
1047 are quite important for correct removal as a strong cue gives away the presence of the object which
1048 makes the object removal very hard or impossible. For these reasons, an end-to-end vision-based
1049 solution is highly desirable and presents a promising direction for future work.