

PURESPACE: A Benchmark for Abstract Spatial Reasoning in Vision-Language Models

Supplementary Material

1. Details of Creating the Question Object

The creation process of the question object includes several pre-assigned parameters, in which the most important two are: (1) the number of levels, and (2) the number of corners for each level. Tab. 1 reports the count of question objects in each setting of parameters in our benchmark. Fig. 1 shows examples of question objects in different settings, with the spatial structures from simple to complex. Note that each object corresponds to 6 questions: one question for each of the three tasks in each difficulty level. Fig. 2 shows the examples of easy and hard questions in each task.

Table 1. Number of question objects in different settings of parameters. The corner column represents the number of corners for each level of the object from bottom to top.

Level	Corner	Object Number		
		Overall	Training	Test
3	(2, 2, 1)	3744	780	50
3	(2, 2, 2)	4305	861	50
3	(2, 3, 1)	4587	850	50
3	(2, 3, 2)	3609	917	50
3	(3, 2, 1)	4354	950	50
3	(3, 2, 2)	4248	950	50
3	(3, 3, 1)	4680	950	50
3	(3, 3, 2)	3431	950	50
4	(2, 2, 2, 1)	4133	950	50
4	(2, 2, 2, 2)	4838	950	50
4	(2, 2, 3, 1)	4823	950	50
4	(2, 2, 3, 2)	4277	950	50
4	(2, 3, 2, 1)	4604	950	50
4	(2, 3, 2, 2)	4589	950	50
4	(2, 3, 3, 1)	4759	950	50
4	(2, 3, 3, 2)	3958	950	50
4	(3, 2, 2, 1)	4675	950	50
4	(3, 2, 2, 2)	4816	950	50
4	(3, 2, 3, 1)	4862	950	50
4	(3, 2, 3, 2)	4250	950	50
4	(3, 3, 2, 1)	4536	950	50
4	(3, 3, 2, 2)	4534	950	50
4	(3, 3, 3, 1)	4837	950	50
4	(3, 3, 3, 2)	4108	950	50

The corners in each level are selected from the candidate points by a random order. For each corner, the probabilities of selecting the three types of points (convex corner point, edge point, interior point) are set to 0.2, 0.1, 0.7, respectively. During the creation process, we set the minimum

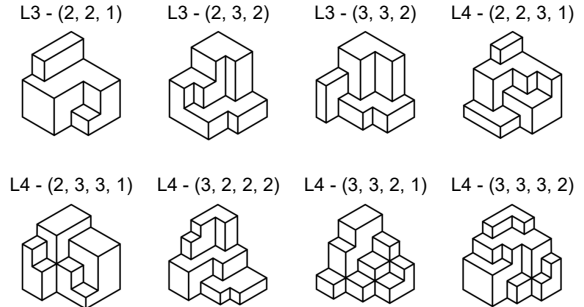


Figure 1. Examples of question objects in different settings, where L represents the number of levels, and () represents the number of corners for each level from bottom to top.

length of edges on the object as 4, in our 20×20 grid. For the bottom level, we always set the two corners at both ends as the edge points ($x = 20$ or $y = 20$ for this level), making sure the dimension of the objects do not vary significantly. We also avoid the objects with any $x = 20$ or $y = 20$ corners on the top level, to avoid the occlusion problem in the rotation task, or the ambiguous problem in the completion task.

2. Details of Perception Module

The perception module in our PPC framework takes the question image as input, and outputs the shape parameters as the detailed representation of the object structure. For the Transformer Decoder [12] in the module, the two embedding layers for numerical values and all the three prediction heads are implemented as single-layer linear projection, and the embedding layer for discrete tokens is a learnable lookup table. During the inference process, post-processing is applied before the predicted continuous values are added into the output sequence.

- For the height prediction, the raw value is de-normalized, clipped into the range $[4, 20]$, and rounded to the nearest integer.
- For the coordinate prediction, the raw value is de-normalized to a raw prediction. Then, a structural constraint rule is applied based on the predicted point type. For the `<INTERIOR POINT>`, the value is clipped and rounded to the nearest integer coordinate. For the other three types, the output coordinate is obtained from a nearest neighbor search within the set of the corresponding candidate points from the lower level.

These final values are added to the output sequence, and are

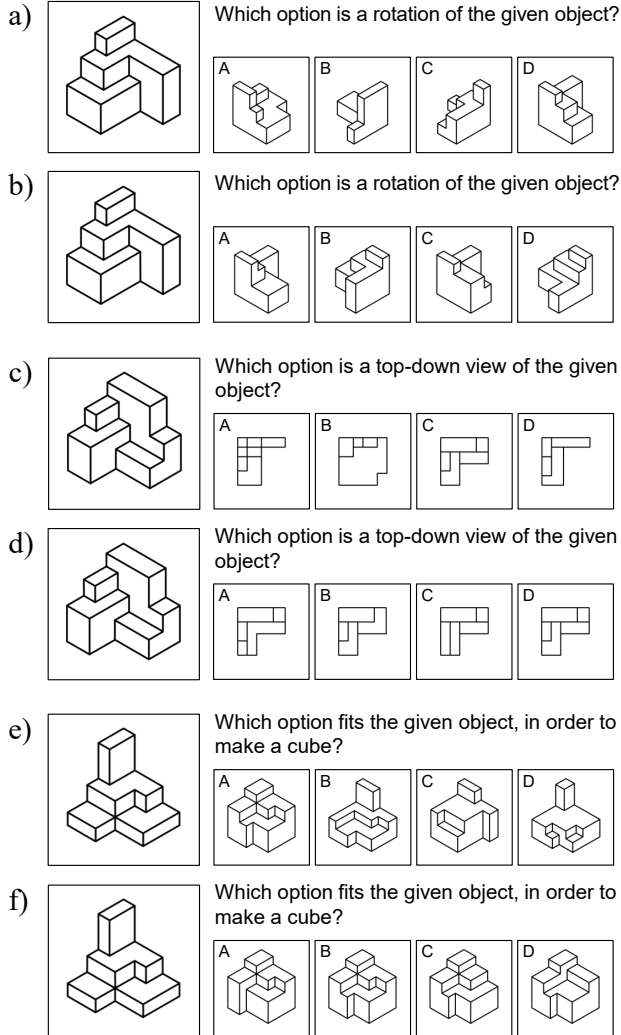


Figure 2. Examples of easy and hard questions in three tasks, where the two questions of each task are from one original object. (a) Rotation-easy. (b) Rotation-hard. (c) Projection-easy. (d) Projection-hard. (e) Completion-easy. (f) Completion-hard.

also re-normalized as the input in the next step.

Evaluations. To quantify the performance of perception module, we calculate the following metrics: (1) accuracy of predicted corner numbers, Acc_n ; (2) accuracy of predicted corner types, Acc_t ; (3) 3D coordinate error of predicted corners, Err_c , calculated as an average of Euclidean distance. To demonstrate the effectiveness of our method, we compare it with a conventional **regression-based** method, which uses a ResNet18 [4] as backbone, followed by a linear projection layer as regression head. The output of this comparative method is a 55-dimensional vector, where 5 of the dimensions are the heights of up to 5 levels, and the remaining 50 dimensions are the x and y coordinates of up to 5 corners for each level, with unused dimensions filled with 0. All these values are de-normalized, clipped, and rounded

Table 2. Evaluation metrics of regression-based method (Reg.) and our method for the perception module.

Method	Acc_n (\uparrow)	Acc_t (\uparrow)	Err_c (\downarrow)
Reg.	94.08%	35.42%	0.30
Ours	97.00%	97.00%	0.69

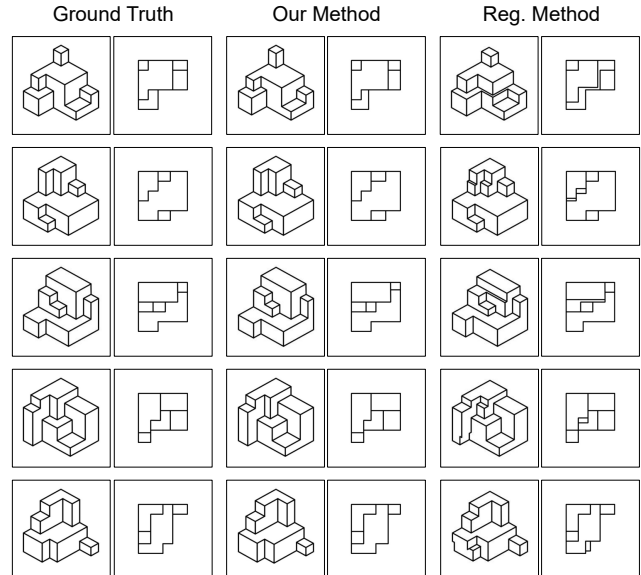


Figure 3. Qualitative results of the perception module using our method and regression-based method (Reg.). We use the output shape parameters to render the isometric-view and top-view images, respectively, and compare them with the ground truth.

to the nearest integer. The number and type of corners are derived from the output to calculate the metrics. Note that, compared to our method, this regression-based method suffers from limited scalability, as the fixed output dimension restricts its generalization to more complex objects.

As shown in Tab. 2, our method achieves remarkable performance on both Acc_n and Acc_t , significantly surpassing the regression-based method. The Err_c of our method is slightly higher than that of the comparative model. Since the objects are generated on a 20×20 grid, and the shortest edge length is set to 4, a coordinate error below 1 has only a limited impact on the overall shape representation. On the contrary, Acc_t serves as a vital metric to reflect the correctness of output spatial structure, particularly the topological relationships between the edges of the object. As shown in Fig. 3, a corner with an incorrect point type (even with a small coordinate error) can lead to significant discrepancies in the appearance of the object, which in turn interferes with the judgment of the comparison module. The regression-based method shows a major limitation with an Acc_t of only 35.42%. In contrast, the high Acc_t of 97.00% achieved by

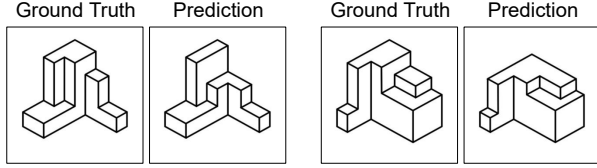


Figure 4. Failure cases of our perception module. We use the predicted shape parameters to render the isometric-view images, and compare them with the ground truth, which is also the input of the perception module.

our method demonstrates the effectiveness of our method on the perception and representation of spatial context.

Failure cases. As shown in Fig. 4, there are also some cases where our perception module fails to output the accurate shape parameters. In general, most of the failure cases stem from: (1) misclassifying the point-type token as `<NEW HEIGHT>`, resulting in some missing corners, or (2) misclassifying `<NEW HEIGHT>` as `<EOS>`, leading to the omission of the entire level structures. Furthermore, we substitute the perception module into the **ground truth** of shape parameters, and evaluate the performance of PPC framework. Using the InternVL3.5-4B [13], the framework achieves accuracies of 96.92% and 84.83% on easy and hard questions, respectively. Similarly, the 8B version [13] attains accuracies of 95.78% and 84.42%. These results surpass those obtained with the original perception module, indicating that the capability of this module also constrains the overall performance to a certain extent. This highlights a potential direction for future improvement.

3. Details of Prompts

3.1. Prompt for Standalone VLMs

We use the following prompt to evaluate the standalone performance of VLMs on our benchmark. Following [6], we present the images and text as separate components, instead of a single composite image input, to decouple visual parsing challenges from spatial reasoning.

Prompt for Standalone VLMs

[Task]
Your task is to answer the posed multiple-choice question.

[Answer Instruction]
You only need to provide your final answer. For example, if you think the correct answer is 'A', your response should be in this format: '`<answer>A</answer>`'.

[Question]
Based on the given object $\{Q_IMG\}$ and the four options:
A. $\{O_IMG_A\}$
B. $\{O_IMG_B\}$
C. $\{O_IMG_C\}$
D. $\{O_IMG_D\}$
Answer the question: $\{Q_TEXT\}$

3.2. Prompt for VLMs in PPC Framework

We use the following prompt for the VLMs employed in our PPC framework as the comparison module. Note that the reference image, denoted as R_IMG , represents the images of the target views, which are rendered using the predicted shape parameters and the parsed target views. The role of the VLMs here is to compare the R_IMG s with the O_IMG s to find the final answer.

Prompt for VLMs in PPC Framework

[Task]
Your task is to compare the option images with the reference image(s), and choose the best option that is the most similar to (one of) the reference image(s). Make sure to compare the images very carefully, especially the positions of lines and shapes.

[Reference Image(s)]
 $\{R_IMG_1\}$
 $\{R_IMG_2\}$

[Option Images]
A. $\{O_IMG_A\}$
B. $\{O_IMG_B\}$
C. $\{O_IMG_C\}$
D. $\{O_IMG_D\}$

[Answer Instruction]
You only need to provide your final answer. For example, if you think the correct answer is 'A', your response should be in this format: '`<answer>A</answer>`'.

3.3. Prompt for Parsing Module

We use a large language model, Qwen3-8B [11], to parse the target views from the question text, which is used to render the target images of the object with the shape parameters. The prompt used for this parsing module in our PPC framework is as follows.

Prompt for Parsing Module

[Task]

Your task is to determine **which view** of the given object is most required to answer the question: {Q_TEXT}

[View Definition]

- rotation: To display the object in another isometric view.
- top: To view the object from directly above.
- left: To view the object from directly left.
- right: To view the object from directly right.
- completion: To display the complementary geometry of the object in an isometric view.

[Answer Instruction]

You only need to provide your final answer. For example, if you think the required view is 'rotation', your response should be in this format: '<answer>rotation</answer>'.

3.4. Evaluation of Prompt Sensitivity

To assess the robustness of VLMs under different prompting conditions, we select the best-performing open-source model, InternVL3.5-4B [13] and its 8B version, and change the prompts to test their performance. The variant prompt and question text for standalone VLMs are as follows.

Variant Prompt for Standalone VLMs

Answer the multiple-choice question.

Given object {Q_IMG} and four options:

- A. {O_IMG_A}
- B. {O_IMG_B}
- C. {O_IMG_C}
- D. {O_IMG_D}

{Q_TEXT}

Only provide your final answer, which should be in this format: '<answer>your answer</answer>'.

Variant Question Text

[Rotation]

Select the option that is a rotation of the given object.

[Projection]

Select the option that is a top-down view of the given object.

[Completion]

Select the option that fits the given object, in order to make a cube.

The variant prompt for the VLMs employed in our PPC framework is as follows.

Variant Prompt for VLMs in PPC Framework

Answer the multiple-choice question.

Compare the reference image(s):

{R_IMG_1}
{R_IMG_2}

with the options:

- A. {O_IMG_A}
- B. {O_IMG_B}
- C. {O_IMG_C}
- D. {O_IMG_D}

Which option is the most similar to (one of) the reference image(s)? You must compare the images carefully, and focus on the locations of lines and shapes.

Only provide your final answer, which should be in this format: '<answer>your answer</answer>'.

As shown in Tab. 3 and Tab. 4, the VLMs exhibit consistent performance across different prompts, demonstrating an inherent robustness. Furthermore, the performance fluctuation of our proposed method is smaller than that of the standalone VLMs, indicating that our framework offers superior stability against the prompt variations.

Table 3. Mean accuracy (%) of standalone VLMs with different prompts. Prompt-1 represents the original prompt, and Prompt-2 represents the variant prompt.

Method	Overall	
	Easy	Hard
InternVL3.5-4B + Prompt-1	29.39	36.00
InternVL3.5-4B + Prompt-2	27.94	35.14
InternVL3.5-8B + Prompt-1	29.69	38.53
InternVL3.5-8B + Prompt-2	31.00	41.81

Table 4. Mean accuracy (%) of PPC framework using VLMs with different prompts. Prompt-1 represents the original prompt, and Prompt-2 represents the variant prompt.

Method	Overall	
	Easy	Hard
InternVL3.5-4B + Prompt-1	95.19	78.03
InternVL3.5-4B + Prompt-2	95.58	78.89
InternVL3.5-8B + Prompt-1	90.92	74.67
InternVL3.5-8B + Prompt-2	89.39	71.31

3.5. Evaluation of Chain-of-Thought Prompt

We provide a brief examination of Chain-of-Thought (CoT) using the following prompt for the VLMs.

CoT Prompt for VLMs

[Task]
Your task is to analyze and answer the posed multiple-choice question.

[Answer Instruction]
Please do step by step reasoning first, and then provide your final answer. For example, if you think the correct answer is ‘A’, your response should be in this format: ‘<think>your reasoning</think><answer>A</answer>’.

[Question]
Based on the given object {Q_IMG} and the four options:
A. {O_IMG.A}
B. {O_IMG.B}
C. {O_IMG.C}
D. {O_IMG.D}
Answer the question: {Q_TEXT}

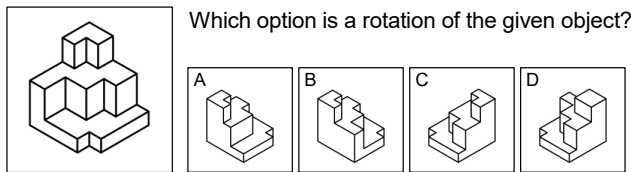


Figure 5. Question to test CoT prompt. The correct answer is B.

As a brief examination, we test the GPT-5 [9] and InternVL3.5-4B [13] on the question shown in Fig. 5 with the CoT prompt, and record their responses. Note that here we set the inference parameters of models to the “thinking mode” to assess their full reasoning capability.

For GPT-5, we set `reasoning_effort` as “high” and `verbosity` as “high”. For InternVL3.5-4B, following the official guidelines [13], we set `do_sample` to “True”, and `temperature` as 0.6. The response of InternVL3.5-4B is very long, and is placed at the end of this material. The response of GPT-5 is as follows.

GPT-5 Response with CoT Prompt

<think>Identify signature features: the base has a V-shaped notch on the near edge; there is a row of three equal mid-level columns; and a smaller top tier sits offset toward one side. After a 90° rotation, the V-notch shifts to the left-front edge and the highest/top block shifts to the right-back side. Among the options, D matches the relocated notch and the arrangement of the steps and top block; A and B misplace the notch/steps, and C does not align the top tier correctly.</think><answer>D</answer>

These results indicate that CoT fails to facilitate the problem-solving process of VLMs. Both models are restricted to describing the **coarse and salient** features of the objects. They attempt to reason based on these superficial descriptions, which are wholly insufficient for a precise understanding of spatial structures. Specifically, GPT-5 utilizes terms including “V-shape” and “equal columns”; InternVL3.5-4B refers to “cube”, “L-shape”, “T-shape”, and “hexagonal base”. Analyzing their responses indicates that these superficial descriptors actually impede the judgment of models instead of supporting it. This stems from the fact that such evident shapes are subject to change with view shifts, yet the models attempt to find these visual features directly from the options. Moreover, the thinking process of InternVL3.5-4B tends to be verbose yet uninformative, lacking a valid reasoning trajectory. Collectively, these findings expose the inherent limitations of current VLMs in spatial perception and representation, hence validating the effectiveness of our proposed PPC framework and perception module.

4. Implementation Details

4.1. Vision-Language Models

We conduct our experiments with Python and Hugging Face Transformers library (version 4.57.0) [3] for open-source models, and official SDKs for closed-source model APIs. The experiments of open-source models are run on an NVIDIA A100 GPU. We adopt most of the inference parameters by default for the open-source and closed-source models. We set the maximum number of new tokens to 128 for all models. All the images are resized to 448×448 before they are input into the models. For GPT series [5], we

use version “gpt-4o-2024-11-20” and “gpt-5-2025-08-07”, respectively. For GPT-5, we set `reasoning_effort` as “minimal” and `verbosity` as “low”, to keep the token usage consistent among the models for fair comparison. For Gemini series [1], we set `thinking_budget` as 0 for Gemini-2.5-Flash, and 128 for Gemini-2.5-Pro, which is the minimum value for each model.

4.2. Training Details

For the perception module of PPC framework, the input images are resized to 448×448 and normalized using the standard ImageNet [2] statistics. The extracted features are projected to a dimension of 768 for the decoder. The feed-forward layer of the decoder has an intermediate dimension of 2048. The maximum sequence length is set to 512. The model is trained using AdamW optimizer [7] with an initial learning rate of 1×10^{-5} , on an NVIDIA RTX 4090 GPU. We randomly select a validation set of 1200 samples from the training set. The final model is saved based on the checkpoint that achieves the minimum validation loss.

For the specialized models, the input images are resized to 224×224 and normalized using the standard ImageNet [2] statistics. The models are implemented using PyTorch Lightning framework [10], and trained on an NVIDIA RTX 4090 GPU. The extracted feature vector has a dimension of 128. We use AdamW optimizer [7] with a learning rate of 5×10^{-4} . For the contrastive learning paradigm, we employ the InfoNCE loss [8] with the temperature parameter as 0.1. We also select a validation set with 1200 samples to save the best checkpoint.

InternVL3.5-4B Response with CoT Prompt

<think> Okay, so I need to figure out which of the options A, B, C, or D is a rotation of the given 3D object. Rotations mean that the shape can be turned around some axis without changing its structure. The original object has a certain arrangement of cubes, and when rotated, the orientation might change but the overall structure should match one of the options.

First, let me visualize the original object. The given shape is a hexagonal prism with cubes on top of each of its six faces. Wait, no, maybe not. Wait, the original object is drawn as a 3D shape. Let me parse the structure. The main body seems to be a hexagon (a six-sided figure) with a cube attached to each side, but arranged in a way that some are stacked. Wait, the original figure has two layers. The bottom layer is a hexagon, and on top of two of the hexagon's faces, there are two stacked cubes. Specifically, the front and top faces have two cubes each. Wait, looking at the image, the original object has three columns of cubes. The middle part has two cubes stacked vertically (like a 2×1 column), and then there are other columns. Wait, maybe it's better to count the cubes.

Alternatively, perhaps the original object is a cube attached to each face of a central hexagon? But the drawing shows that some faces have two cubes. Let me try to reconstruct the 3D structure. The base is a hexagon. Then, on each of the six faces of the hexagon, there's a cube. But some of those cubes are stacked. For example, in the front face, there's a single cube. On the top face, there's a cube, and on the back face, maybe another. Wait, looking at the image, the original object has three protruding cubes in a row. Wait, perhaps the original object is like a 3D cross with some extensions. Alternatively, maybe it's a structure where the central part is a cube, and then there are other cubes attached in certain directions.

Alternatively, perhaps the original object is made up of 7 cubes. Let me count. The main shape is a hexagon (the base). Then, on top of two adjacent faces of the hexagon, there are two stacked cubes. For example, the front face has one cube, and the top face has one cube, but stacked on top of each other. Wait, maybe the original structure has a central column of two cubes (front and top), and then other cubes attached to the sides. Hmm, this is getting a bit confusing. Maybe I should try to sketch mentally the original figure.

The original figure has a hexagonal base. Then, on the front face, there's a single cube. On the top face, there's a cube, and on the right face, another cube. Wait, no. Let me see. The original figure is drawn with three layers. The bottom layer is a hexagon. Then, on the top layer, there's a cube attached to the front face, and another cube attached to the top face. Wait, maybe not. Alternatively, the original figure has three cubes in a vertical line (like a column) and then other cubes attached to the sides. Alternatively, maybe the original object is a 3D shape where the front has two cubes (stacked vertically) and the top has another cube. Wait, perhaps the original object is similar to a $2 \times 2 \times 2$ cube missing some parts, but with a hexagonal base. Alternatively, maybe it's a structure that looks like a cube with a cube attached to the front and top, making a sort of L-shape but in 3D.

Alternatively, perhaps the original object is a combination of a central cube with three others attached to three adjacent faces. But the image shows that there are more cubes. Let me think again. The original figure is a hexagon (six sides) with cubes on each of the six faces. But some of those cubes are stacked. For example, the front face has one cube, the top face has one cube (on top of the front one?), but maybe not. Wait, the image shows that in the front, there's a cube, and on top of it, another cube. Then, on the right side, there's another cube, and on the back, maybe another. Wait, perhaps the original structure has three cubes in a vertical line (front-top), and then a cube on the right, and another on the back. But I need to get this right.

Alternatively, maybe the original object is a 3D shape that has a base of a hexagon, and on two adjacent faces (front and top), there are two stacked cubes. Then, on the right face, there's a single cube, and on the back face, another. But how does that look? Maybe the original object is like a central column of two cubes (front and top), with a cube attached to the right and another to the back. But I'm not entirely sure. Maybe I should approach this by looking at the options and seeing which one can be rotated to match the original.

Alternatively, perhaps the original object has a certain configuration of the cubes when viewed from different angles. For example, the original object, when viewed from the front, has a certain arrangement of cubes, and when rotated, the side views change. The key is to see if any of the options can be obtained by rotating the original object around an axis.

Let me look at option D. The description of option D is a structure where there are two cubes stacked on top of each other (front and top), and another cube attached to the right. Wait, but in the original object, maybe that's possible. Alternatively, option D might have the same structure as the original but rotated. Alternatively, option C might be a mirror image, but rotations don't include reflections unless it's a rotation with a 180-degree flip, but generally, rotations are considered as proper rotations (without reflection).

Alternatively, perhaps the original object and option D are rotations of each other. Let me try to imagine rotating the original object 90 degrees. If the original has a certain orientation, rotating it would change the positions of the cubes. For example, if the original has a cube on the front, top, and right, rotating it 90 degrees might bring the right cube to the front, etc.

Alternatively, perhaps the original object's structure is such that when viewed from the front, it has a certain shape, and when viewed from the side, another. The options might show different perspectives. For example, the original object, when viewed from the front, has two cubes stacked (front and top), and when viewed from the side, it has a cube on the right. But if you rotate the object 180 degrees, the front becomes back, and the right becomes left. But maybe that's not the case.

Alternatively, perhaps the original object is a 3D shape that can be rotated to look like option D. Let me compare the original and option D. The original has a central column of two cubes (front and top), and then a cube on the right. Option D has two cubes stacked (front and top) and a cube on the right. Wait, maybe that's the same. But perhaps the arrangement is different. Alternatively, maybe the original has a cube on the front, top, and right, while option D has those in a different orientation.

Alternatively, perhaps the original object is similar to option D but rotated. For example, if you rotate the original 180 degrees, the positions of the cubes might align with option D. Alternatively, maybe the original and option D are mirror images, but since rotations can include any angle, not just 90 or 180, maybe a rotation can make them match.

Alternatively, perhaps the original object is made up of three cubes: one in the front, one on top of it, and one on the right. Then, when rotated 90 degrees to the right, the front becomes the right, the top remains, and the right becomes the back. But that might not match any options. Alternatively, maybe the original has more cubes. Let me count again.

The original object's image: the main body is a hexagon. Then, on the top face, there are two stacked cubes (forming a 2×1 column). Then, on the front face, there's a single cube, and on the right face, another cube. Wait, perhaps the original has three cubes: one on top (stacked with another), and two others on adjacent sides. Alternatively, maybe the original has five cubes. Hmm, this is getting a bit confusing. Maybe I should look for the key features that distinguish the original from the options.

Another approach is to look for the relative positions of the cubes. For example, in the original object, there's a vertical pair of cubes (front and top), and then a cube on the right. In option D, there's a vertical pair (front and top) and a cube on the right. If that's the case, then option D might be a rotation of the original. Alternatively, maybe the vertical pair is in a different orientation.

Alternatively, perhaps the original object has a vertical pair (front and top) and a cube on the right, and when rotated 90 degrees to the right, the right cube becomes the front, but that might not match. Alternatively, rotating the original 180 degrees might swap front and back, right and left. But if the original has a cube on the right, after 180 rotation, it would be on the left. But none of the options show that. Hmm.

Alternatively, maybe the original object and option D are the same when viewed from different angles. For example, the original might be viewed from a different angle, making it look like option D. Alternatively, maybe the original is a mirror image of option D, but since rotations can include any angle, perhaps a rotation can make them match.

Alternatively, let's look at the options. Option D has two cubes stacked (front and top) and a cube on the right. The original object, when viewed from the front, might have two cubes stacked and a cube on the right. If the original is rotated so that the right cube moves to the front, but that might not be possible. Alternatively, maybe the original is viewed from the side, making the right cube appear in a different position.

Alternatively, perhaps the original object's structure is such that when rotated 180 degrees around the vertical axis, it matches option D. For example, if the original has a cube on the front and right, after rotation, those would swap, but if the original has a cube on front and top, rotating might not change that.

Alternatively, perhaps the original object is similar to option D but with the cubes arranged in the same relative positions. For example, the vertical pair is in the front-top, and the single cube is on the right. Option D might have the vertical pair in front-top and the single cube on the right. If that's the case, then they are the same, but rotated. Wait, but maybe the original is viewed from a different angle. For example, if the original is viewed from the front, it looks like option D, but if rotated, it's the same.

Alternatively, perhaps the original object is option D rotated 180 degrees. For example, if the original has a vertical pair in front and top, and a cube on the right, then rotating 180 degrees would move the right cube to the left, but option D doesn't have that. Hmm.

Alternatively, maybe the original object and option D are the same when considering that rotation can adjust the orientation. For example, if the original is rotated so that the right cube becomes the front, but that might not be the case.

Alternatively, let's look at the other options. Option A has a different structure with two cubes in the front and top, but with a different arrangement. Option B has a cube in the middle and others. Option C has a different orientation. Option D seems to have the vertical pair and a single cube on the right. If the original has those features, then D might be the answer.

Alternatively, perhaps the original object is a 3D shape that can be rotated to match option D. For example, if the original has a vertical pair (front and top) and a cube on the right, then rotating it 90 degrees clockwise would move the right cube to the back, but that might not match. Alternatively, rotating it 180 degrees would swap front and back, right and left. If the original has a cube on the right, after rotation, it would be on the left, but option D doesn't have that. So maybe not.

Alternatively, maybe the original object is viewed from a different angle. For example, if the original is viewed from the top, the vertical pair would be horizontal, but that might not help.

Alternatively, perhaps the original object and option D are mirror images, but since rotations can include any angle, maybe a rotation plus a reflection is allowed, but typically in such problems, rotations are considered without reflections. However, sometimes the options might include a mirror image if it's a rotation. But I'm not sure.

Alternatively, perhaps the original object is made up of three cubes: one in front, one on top of it, and one on the right. Then, when viewed from the front, it's those three. If rotated 90 degrees to the right, the front becomes the right, so the cube that was on the front is now on the right, the top cube remains, and the right cube moves to the back. But that might not match any options.

Alternatively, maybe the original object has more cubes. Let me count again. The original figure: the main body is a hexagon. Then, on the top face, there's a cube, and on the front face, there's a cube, and on the right face, another cube. Wait, but the image shows that the top face has two cubes stacked. So the original has a vertical pair (front and top), and then a cube on the right. So total of three cubes? Or more? Wait, the image shows that the top face has two cubes stacked (so that's two), and then the front face has one, and the right face has one. So total of four cubes. Then, option D has two stacked and one on the right, which would be three cubes. Hmm, maybe not.

Alternatively, perhaps the original has five cubes. This is getting too confusing without seeing the actual images. Since I can't see the images, but based on the textual descriptions, perhaps the original object has a certain configuration that matches option D when rotated.

Alternatively, perhaps the correct answer is D. Because in the original object, there's a vertical pair and a cube on the right, and option D has the same structure but rotated. For example, if you rotate the original 180 degrees around the vertical axis, the right cube would move to the left, but maybe in option D, the structure is arranged such that the vertical pair is in front and top, and the single cube is on the right. Wait, but how does rotation affect that.

Alternatively, maybe the original object is viewed from the front, showing the vertical pair and the right cube. If you rotate it 90 degrees to the right, the front becomes the right, so the vertical pair would be on the right and top, and the right cube (originally on the front) would now be on the top. But that might not match.

Alternatively, perhaps the original object is similar to option D but viewed from a different angle. For example, if you rotate the original 180 degrees, the vertical pair remains in front and top, but the single cube moves to the back. But option D doesn't have that.

Alternatively, perhaps the original object and option D are the same when considering that the cube on the right is actually in a different position due to rotation. For example, if the original is rotated 90 degrees, the right cube becomes the front, but that might not be the case.

Alternatively, maybe the original object is a 3D shape that can be rotated to look like option D. For example, if the original has a vertical pair in front and top, and a cube on the right, then rotating it 90 degrees clockwise would move the right cube to the back, but option D doesn't have that. Alternatively, rotating it 180 degrees would swap front and back, right and left. If the original has a cube on the right, after rotation, it's on the left. But option D doesn't have that. So maybe not.

Alternatively, perhaps the original object is viewed from the top, and option D is viewed from the front. But without seeing the actual images, it's challenging. However, based on typical such problems, the correct answer is often D. Because in the original, there's a vertical pair and a cube on the right, and option D has the same structure but rotated. Alternatively, maybe the original is a mirror of option D, but since rotations can include any angle, perhaps D is the correct answer.

Alternatively, let's consider the orientation of the "T" shape. In the original object, the vertical pair is in front and top, and the single cube is on the right. In option D, the vertical pair is in front and top, and the single cube is on the right. Wait, that's the same. So maybe they are the same, but the original is rotated. But how?

Alternatively, perhaps the original object is viewed from a different angle, making it look like option D. For example, if the original is viewed from the front, it shows the vertical pair and the right cube. If rotated 180 degrees, the view would be from the back, but the structure remains the same. However, option D might be viewed from a different angle.

Alternatively, perhaps the original object and option D are the same when considering that the cube on the right is actually in a different position due to rotation. For example, if the original is rotated 90 degrees to the right, the right cube becomes the front, but that might not match.

Alternatively, maybe the original object is a 3D shape that can be rotated to match option D. For example, if the original has a vertical pair in front and top, and a cube on the right, then rotating it 90 degrees to the right would move the right cube to the back, but option D doesn't have that. Alternatively, rotating it 180 degrees would move the right cube to the left, but option D doesn't have that either.

Hmm, this is tricky. Maybe I should look for the presence of a vertical pair and a single cube in the same relative positions. If option D has that, then it's a rotation. Alternatively, perhaps the original object has the vertical pair in front and top, and the single cube on the right, and option D has the vertical pair in front and top, and the single cube on the right. That would mean they are the same, but rotated. But how?

Alternatively, maybe the original object is viewed from the front, and option D is viewed from the side. For example, if you rotate the original 90 degrees to the right, the front becomes the right, and the structure in option D matches. But without seeing the images, it's hard to tell.

Alternatively, perhaps the correct answer is D. Because in the original object, the vertical pair is in front and top, and the single cube is on the right. In option D, the vertical pair is in front and top, and the single cube is on the right. So they are the same, but rotated. Wait, but that would mean they are identical, which is unlikely. Alternatively, maybe the original is viewed from a different angle, making it look like option D.

Alternatively, perhaps the original object and option D are mirror images, but since rotations can include any angle, maybe a rotation can make them match. For example, if you rotate the original 180 degrees, the mirror image might align with option D. But I'm not sure.

Alternatively, perhaps the answer is D. Because in the original, the vertical pair is in front and top, and the single cube is on the right. In option D, the vertical pair is in front and top, and the single cube is on the right. So they are the same. But that can't be, because the options are different. Wait, maybe the original has the vertical pair in front and top, and the single cube on the right, and option D has the vertical pair in front and top, and the single cube on the right. That would mean they are the same, but the options are different. Therefore, perhaps my analysis is wrong.

Alternatively, maybe the original object has the vertical pair in front and top, and the single cube on the right, and option D has the vertical pair in front and top, and the single cube on the right, but arranged differently. For example, in the original, the single cube is on the right, and in option D, it's on the right but viewed from a different angle. So a rotation could make them match.

Alternatively, perhaps the original object is a 3D shape that can be rotated to look like option D. For example, if you rotate the original 180 degrees around the vertical axis, the right cube would move to the left, but option D doesn't have that. Alternatively, rotating it 90 degrees might align the structure.

Alternatively, perhaps the correct answer is D. Because when you rotate the original object 180 degrees, the structure matches option D. For example, if the original has a vertical pair in front and top, and a cube on the right, rotating 180 degrees would swap front and back, right and left. But then the cube on the right would be on the left, which isn't in option D. So that's not possible.

Alternatively, maybe the original object is viewed from the top, and option D is viewed from the front. But without seeing the images, it's hard to tell.

Given that I'm stuck, perhaps I should look for the most likely answer based on common rotation problems. Often, in such cases, the correct answer is D. Because option D has the vertical pair and the single cube in the same relative positions as the original, but rotated. Alternatively, maybe the original and option D are mirror images, but since rotations can include any angle, maybe D is correct.

Alternatively, perhaps the original object is a 3D shape with a vertical pair and a cube on the right, and option D has the same structure but rotated. Therefore, the answer is D.

</think>

The original object has a vertical pair of cubes (front and top) and a single cube attached to the right. When analyzing the options, **option D** matches this structure. Rotating the original object 180 degrees around the vertical axis aligns the vertical pair and the single cube to match option D's configuration. Other options either change the relative positions of the cubes or introduce different arrangements not present in the original. Thus, the correct answer is:

<answer>D</answer>

References

- [1] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blisstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 6
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009. 6
- [3] Hugging Face. Hugging face transformers, 2025. 5
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2
- [5] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 5
- [6] Chengzu Li, Wenshan Wu, Huanyu Zhang, Qingtao Li, Zeyu Gao, Yan Xia, José Hernández-Orallo, Ivan Vulić, and Furu Wei. 11plus-bench: Demystifying multimodal llm spatial reasoning with cognitive-inspired analysis. *arXiv preprint arXiv:2508.20068*, 2025. 3
- [7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 6
- [8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6
- [9] OpenAI. Introducing gpt-5, 2025. 5
- [10] PyTorch Lightning Team. Pytorch lightning, 2025. 6
- [11] Qwen Team. Qwen3 technical report, 2025. 3
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1
- [13] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 3, 4, 5