SeeThrough3D
Occlusion Aware 3D Control in Text-to-Image Generation

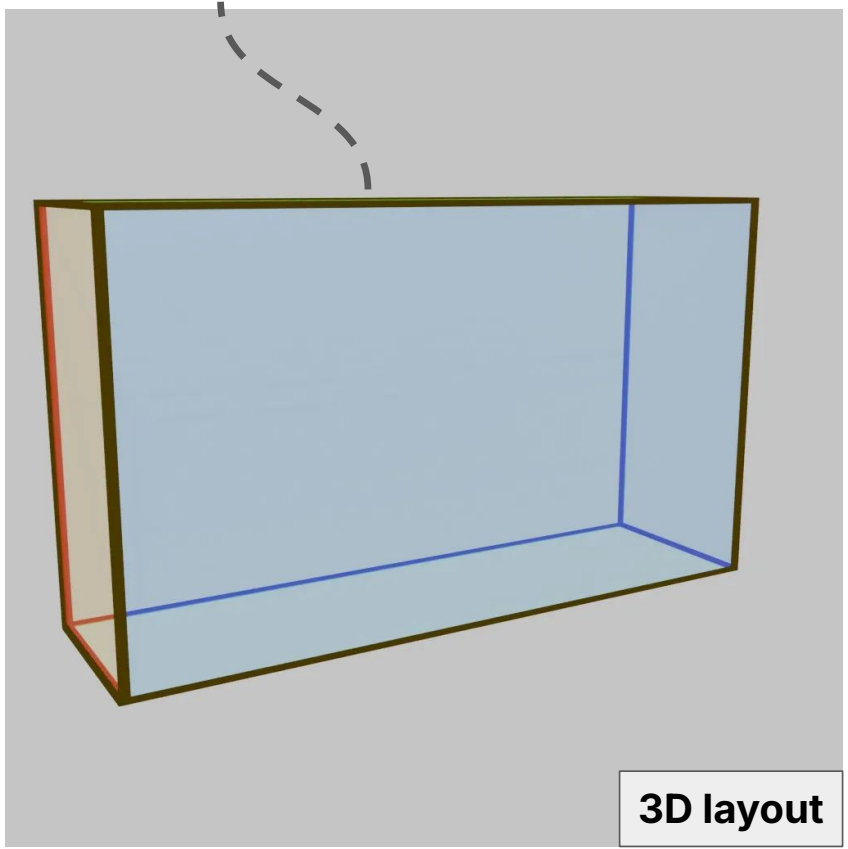
Vaibhav Agrawal Rishubh Parihar Pradhaan S Bhat

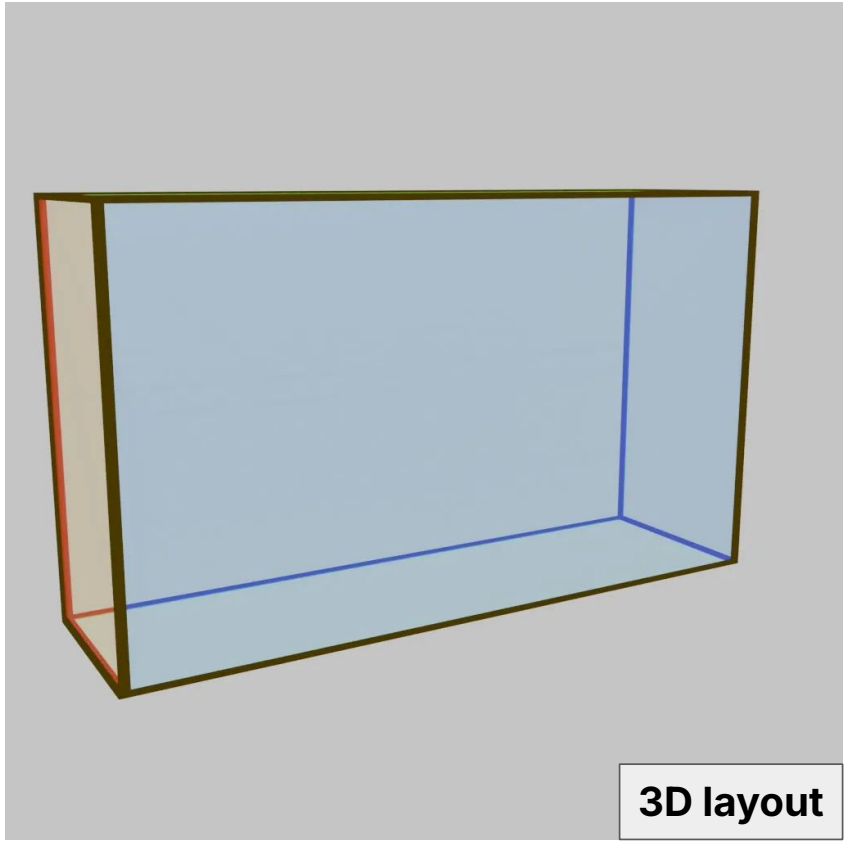
Ravi Kiran S* Venkatesh Babu Radhakrishnan*

* indicates equal advising

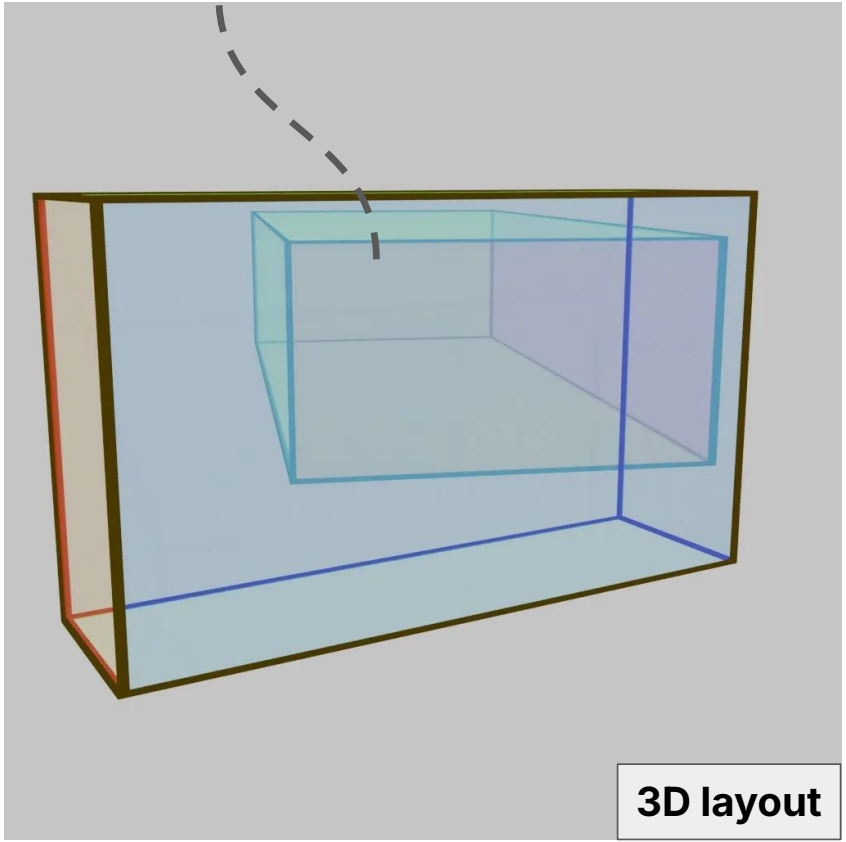
Motivation for occlusion aware 3D control.

*A **bicycle***

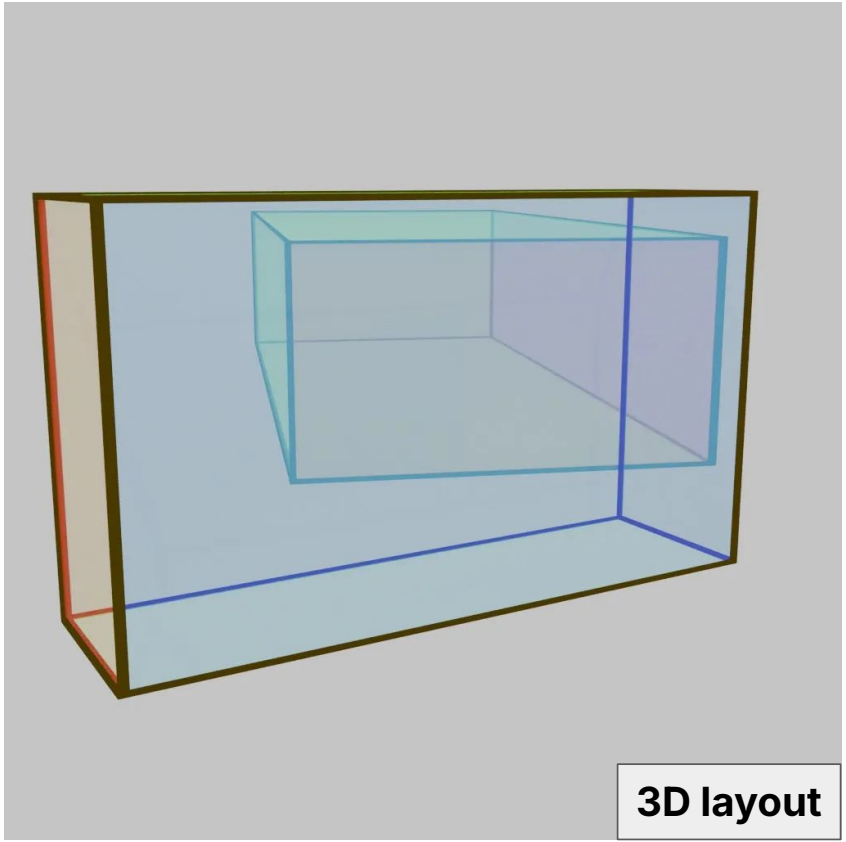



3D layout

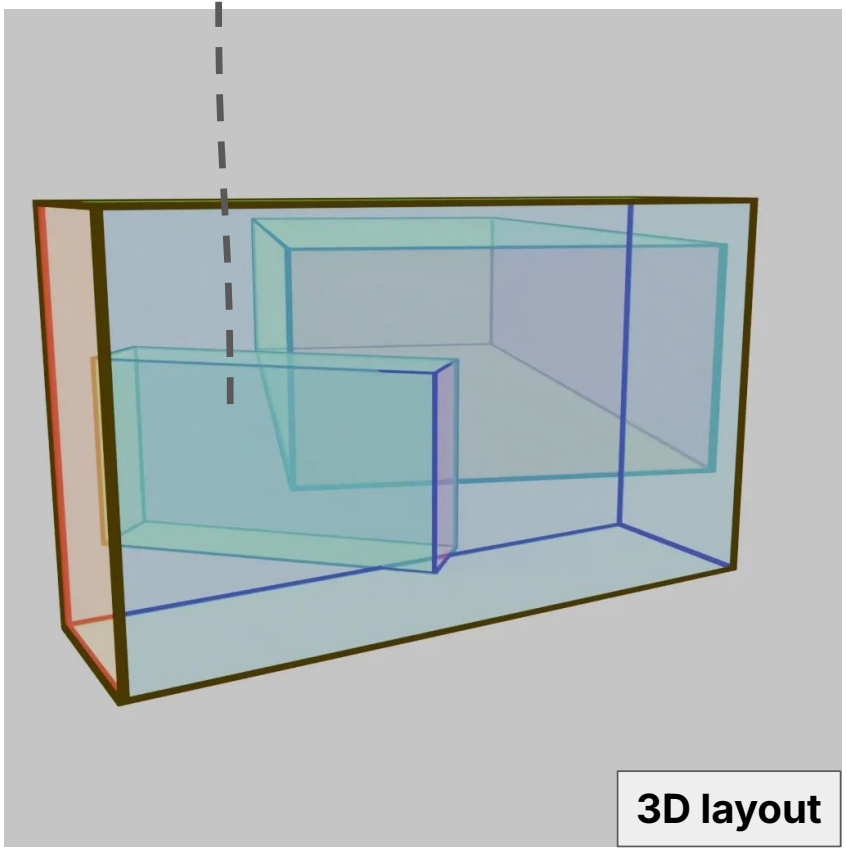
*A **sedan** behind the bicycle*



*High **occlusion** consistency!*

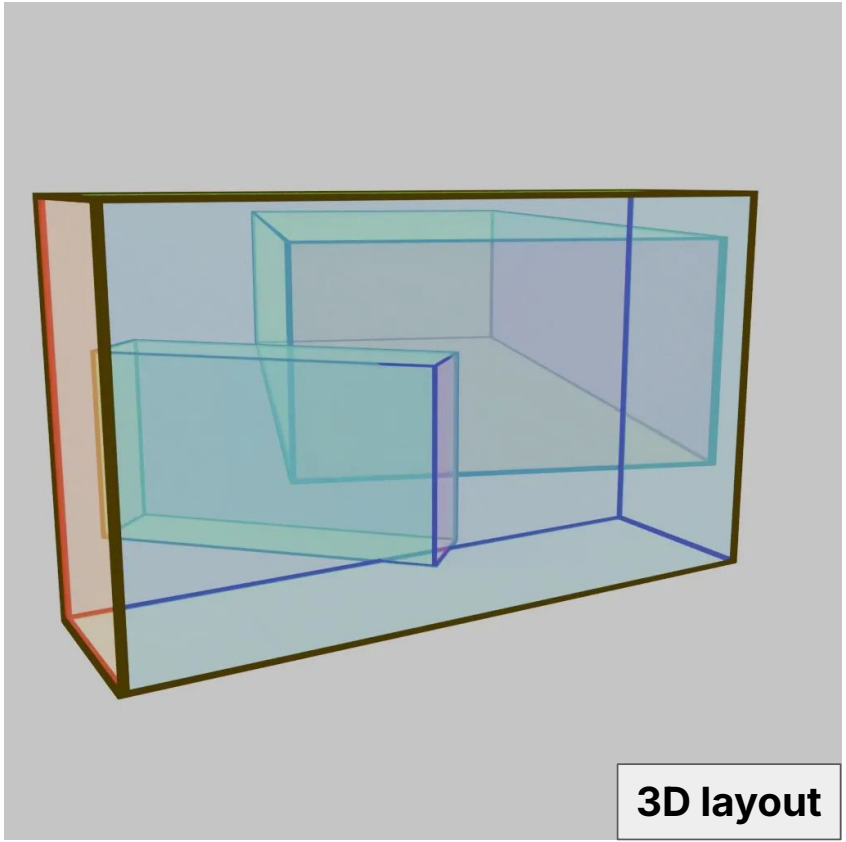

3D layout

*A **dog** behind the bicycle, in front of the sedan*

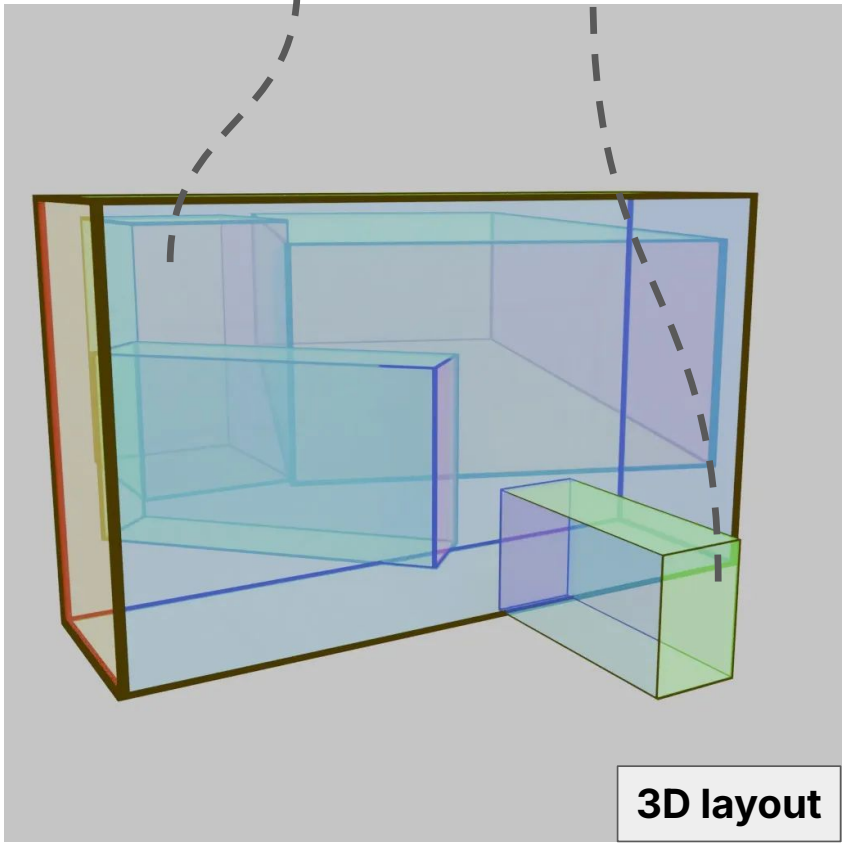


*A **dog** behind the bicycle, in front of the sedan*

*High **occlusion** consistency!*


3D layout

A ***chair****, a* ***puppy***



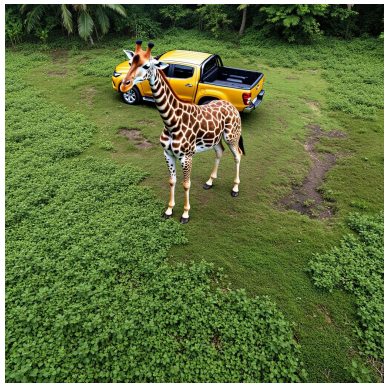
Review. Existing works on 3D layout control

- We observe that occlusion has largely been overlooked in these methods [1,2].

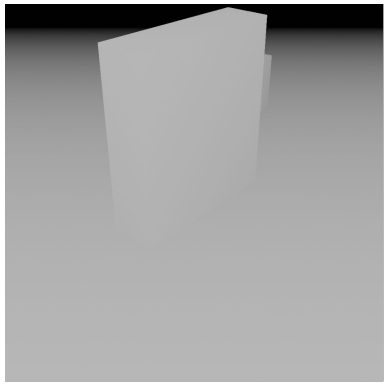
[1] Bhat, Shariq Farooq, Niloy Mitra, and Peter Wonka. "Loosecontrol: Lifting controlnet for generalized depth conditioning." ACM SIGGRAPH 2024.
[2] Wang, Qinghe, et al. "Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation." ACM SIGGRAPH 2025.

Review. Existing works on 3D layout control

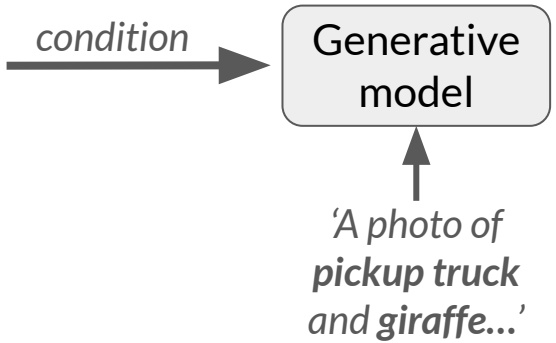
- We observe that occlusion has largely been overlooked in these methods [1,2].
- The scene is typically represented using a ***layout depth map***, which is used to condition the generative model.



Intended scene



Scene representation:
Layout depth map

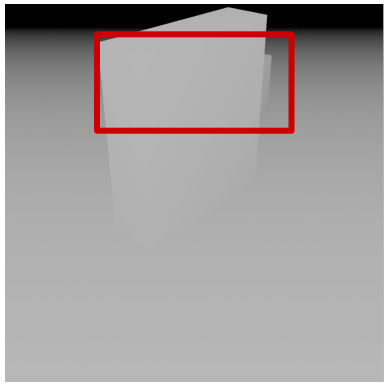


[1] Bhat, Shariq Farooq, Niloy Mitra, and Peter Wonka. "Loosecontrol: Lifting controlnet for generalized depth conditioning." ACM SIGGRAPH 2024.
[2] Wang, Qinghe, et al. "Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation." ACM SIGGRAPH 2025.

Review. Existing works on 3D layout control

- However, layout depth map presents a crucial limitation.

Intended scene



Scene representation:

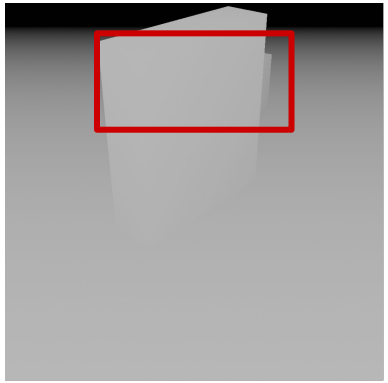
Layout depth map

[1] Bhat, Shariq Farooq, Niloy Mitra, and Peter Wonka. "Loosecontrol: Lifting controlnet for generalized depth conditioning." ACM SIGGRAPH 2024.
[2] Wang, Qinghe, et al. "Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation." ACM SIGGRAPH 2025.

Review. Existing works on 3D layout control

- However, layout depth map presents a crucial limitation.
- It fails to represent **occluded objects.**

Intended scene



Scene representation:

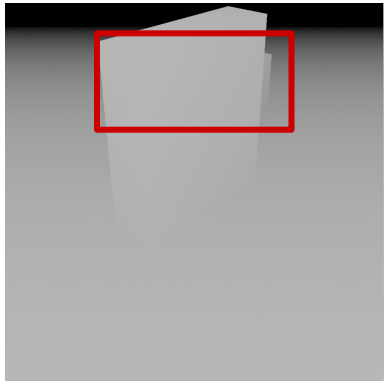
Layout depth map

[1] Bhat, Shariq Farooq, Niloy Mitra, and Peter Wonka. "Loosecontrol: Lifting controlnet for generalized depth conditioning." ACM SIGGRAPH 2024.
[2] Wang, Qinghe, et al. "Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation." ACM SIGGRAPH 2025.

Review. Existing works on 3D layout control

- However, layout depth map presents a crucial limitation.
- It fails to represent **occluded objects.**
- As a result, these models often fail to respect inter-object occlusions.

Intended scene



Scene representation:
Layout depth map

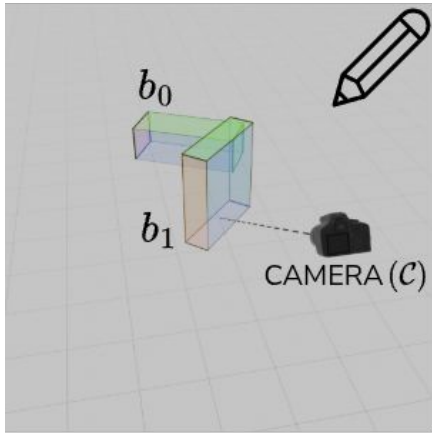
[1] Bhat, Shariq Farooq, Niloy Mitra, and Peter Wonka. "Loosecontrol: Lifting controlnet for generalized depth conditioning." ACM SIGGRAPH 2024.
[2] Wang, Qinghe, et al. "Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation." ACM SIGGRAPH 2025.

Towards occlusion aware representations.

- We propose **OSCR** (**O**cclusion-Aware 3D **Sc**ene **R**epresentation).

Towards occlusion aware representations.

- We propose **OSCR** (**O**cclusion-Aware 3D **Sc**ene **R**epresentation).
- Objects are described using **translucent 3D bounding boxes.**

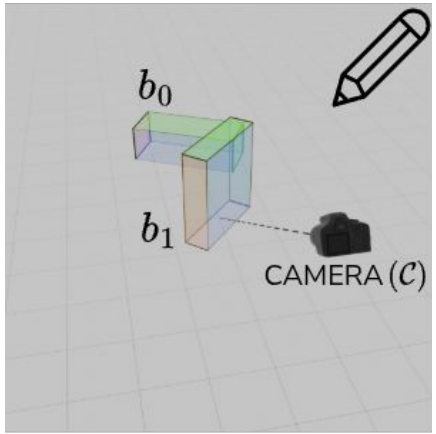


User interface

Transparency captures occluded objects

Towards occlusion aware representations.

- We propose **OSCR** (**O**cclusion-Aware 3D **Sc**ene **R**epresentation).
- Objects are described using **translucent 3D bounding boxes.**
- Faces are **color-coded** to indicate 3D orientation.



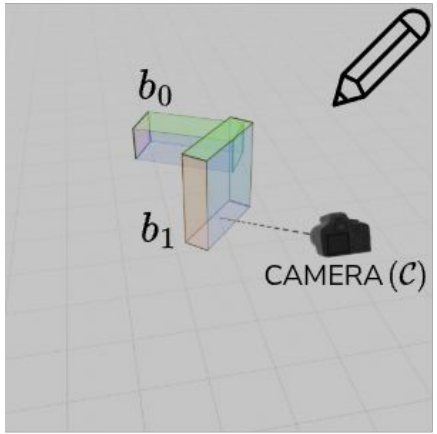
User interface

Transparency captures occluded objects

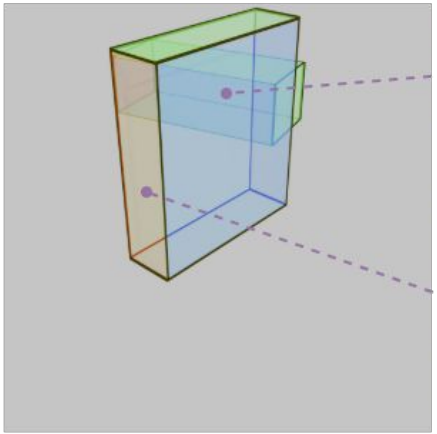
Color-coding faces to indicate 3D orientation (**orange** for front face, **blue** for left, others **green**).

Towards occlusion aware representations.

- We propose **OSCR** (**O**cclusion-Aware 3D **Sc**ene **R**epresentation).
- Objects are described using **translucent 3D bounding boxes.**
- Faces are **color-coded** to indicate 3D orientation.
- Rendered from desired viewpoint to obtain the final representation.



User interface



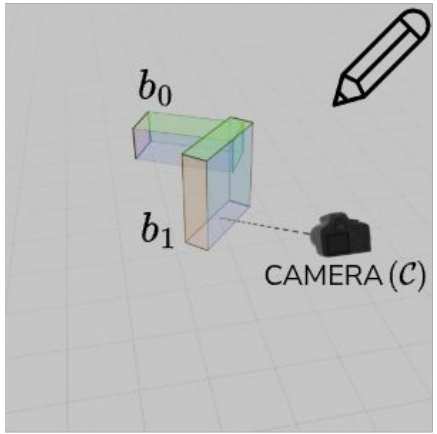
OSCR

Transparency captures occluded objects

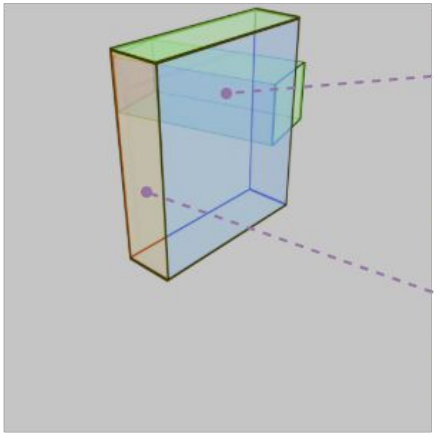
Color-coding faces to indicate 3D orientation (orange for front face, blue for left, others green).

Towards occlusion aware representations.

- We propose **OSCR** (**O**cclusion-Aware 3D **Sc**ene **R**epresentation).
- Objects are described using **translucent 3D bounding boxes.**
- Faces are **color-coded** to indicate 3D orientation.
- Rendered from desired viewpoint to obtain the final representation.



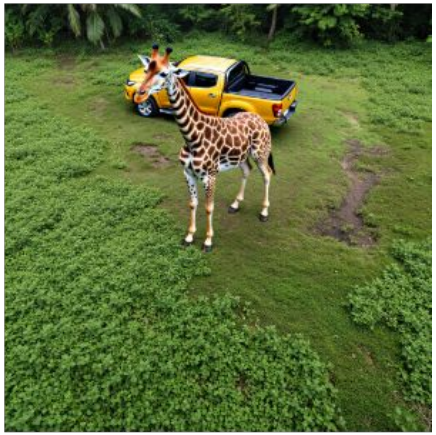
User interface



OSCR

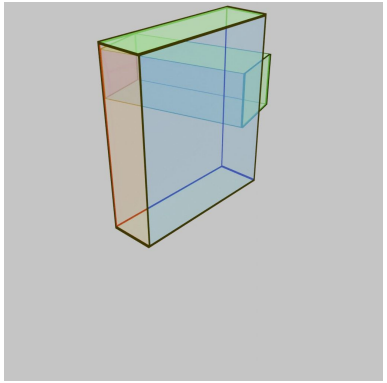
Transparency captures occluded objects

Color-coding faces to indicate 3D orientation (orange for front face, blue for left, others green).



Generated image

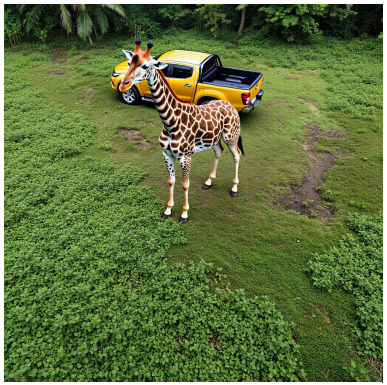
Framework



Rendered OSCR representation

*'A photo of **pickup truck** and **giraffe** amongst vegetation.'*

Text-to-image model (FLUX)



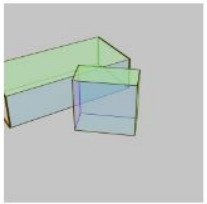
Generated image

Conditioning FLUX with *OSCR*.

'A photo of car and deer in a grassy field'

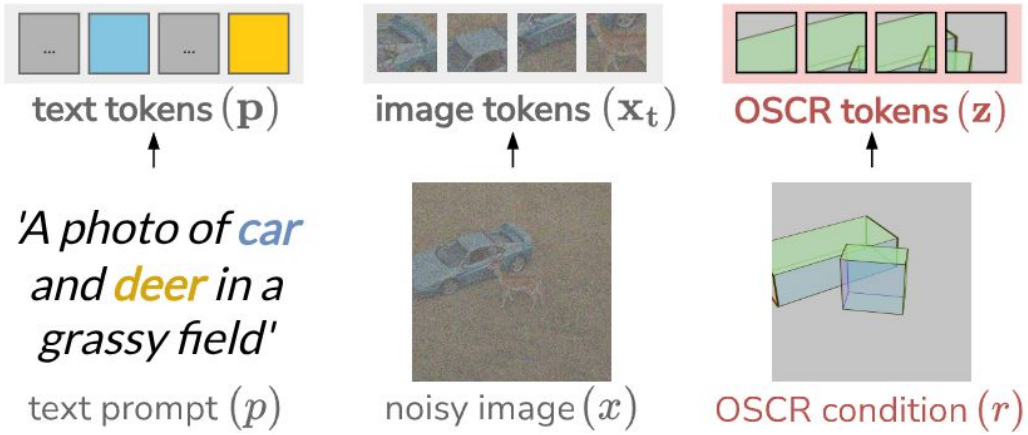
text prompt (p)

noisy image (x)

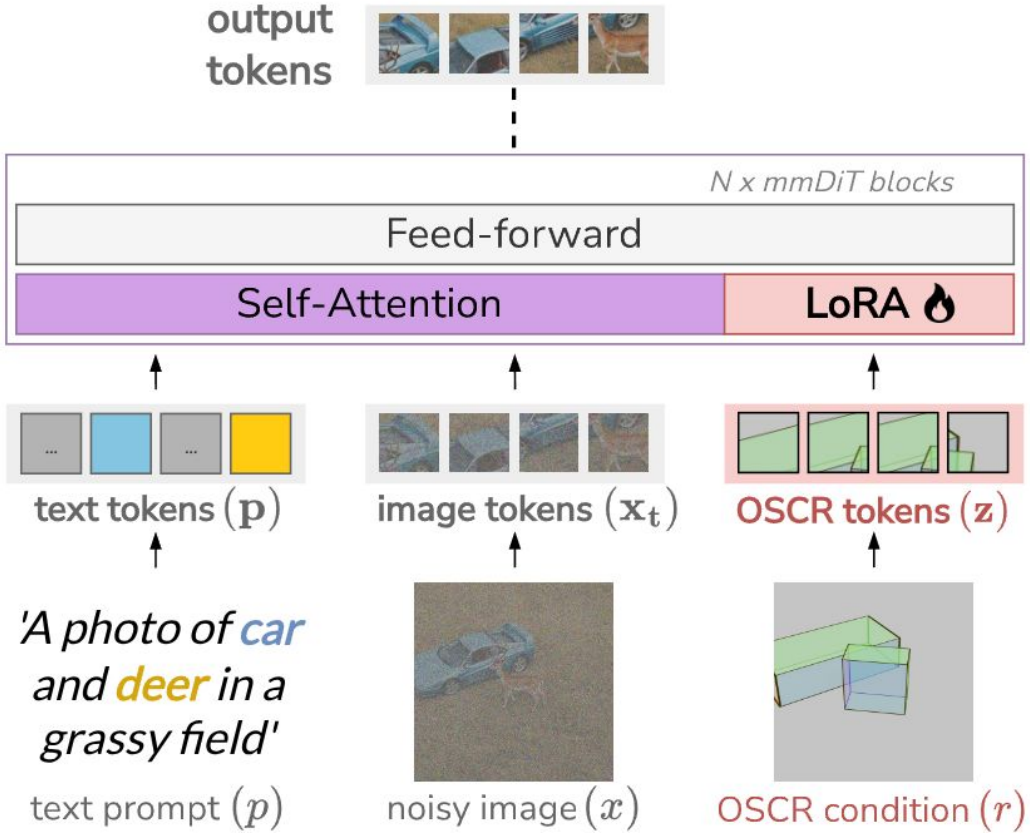


OSCR condition (r)

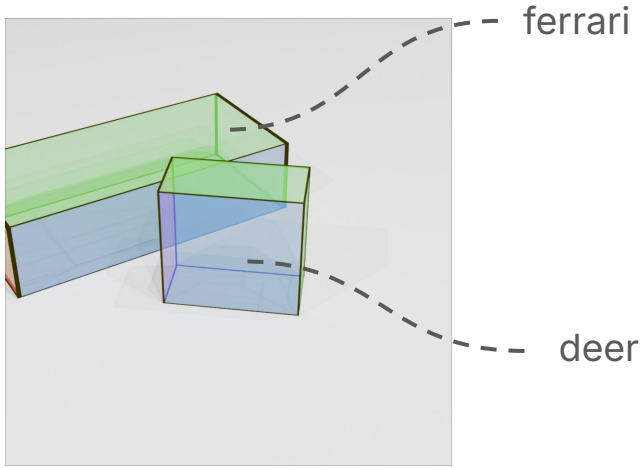
Conditioning FLUX with *OSCR*.



Conditioning FLUX with *OSCR*.

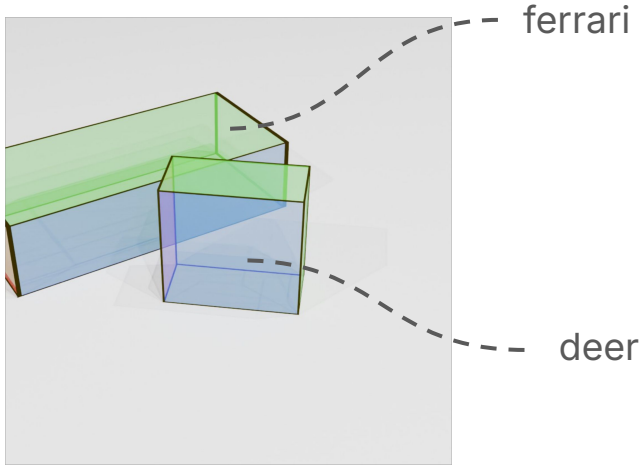


How to bind boxes to corresponding objects?



OSCR condition

How to bind boxes to corresponding objects?

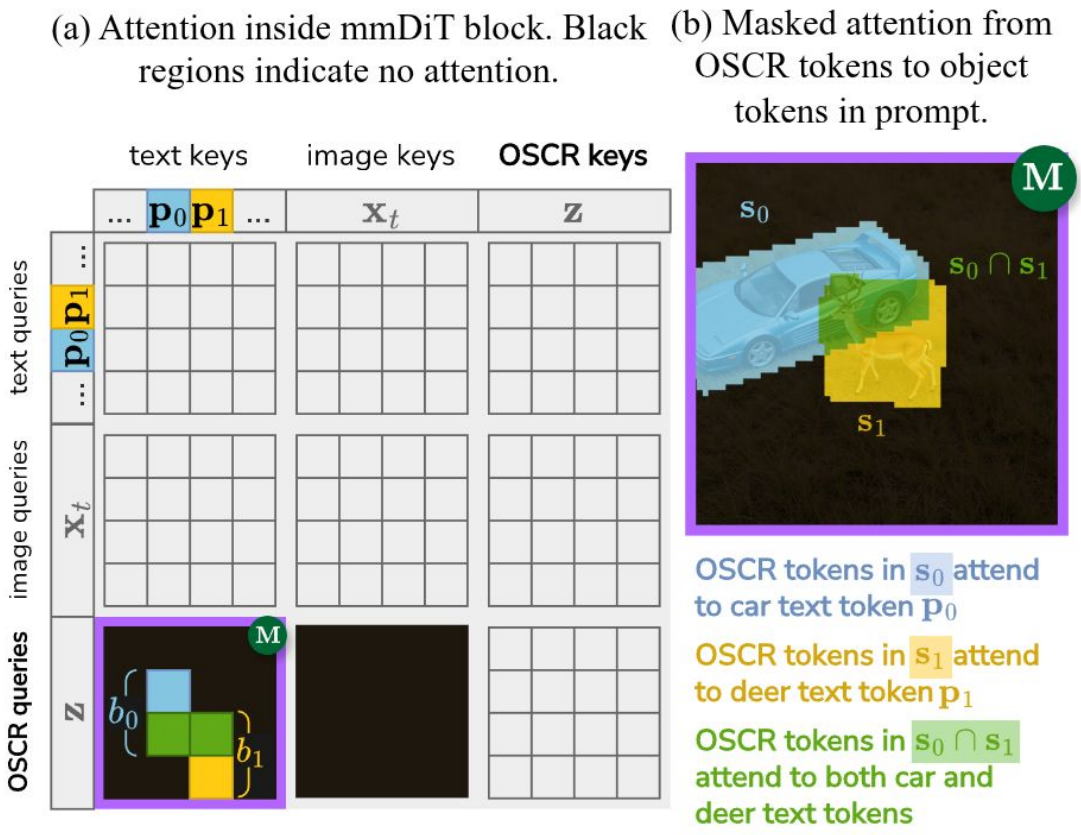


OSCR condition

Correct layout following in image generation

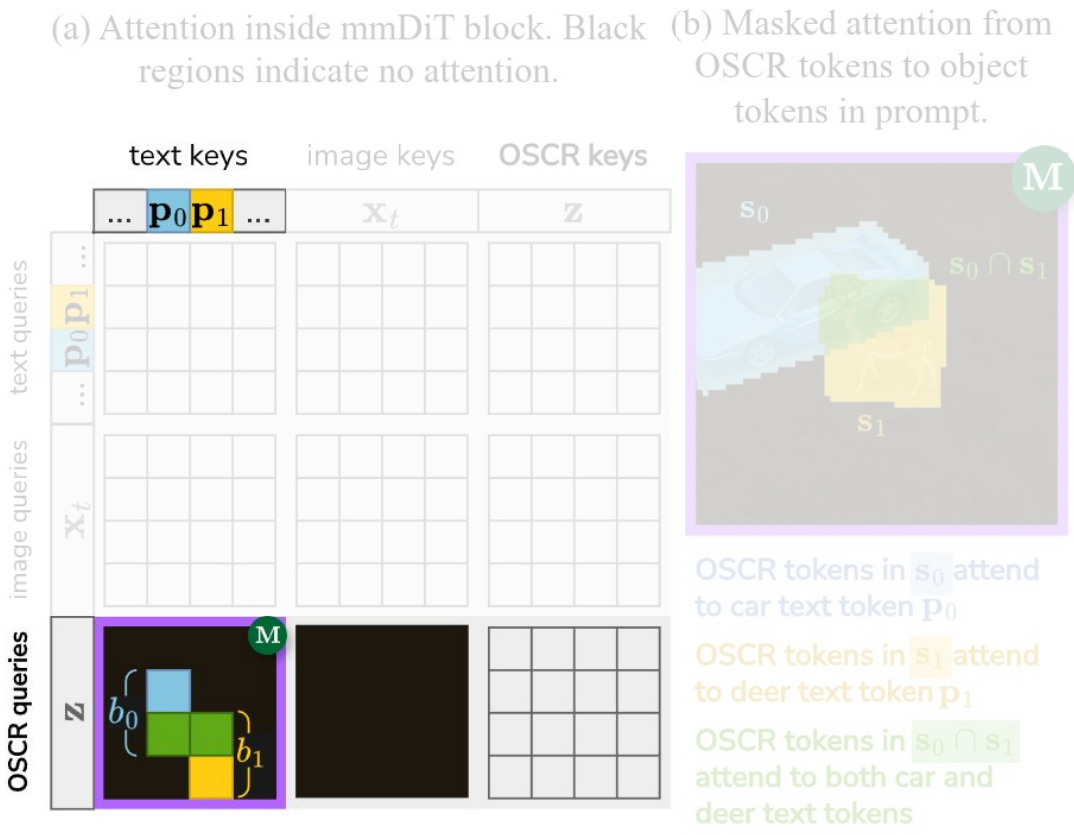
How to bind boxes to corresponding objects?

- **Masked attention!**



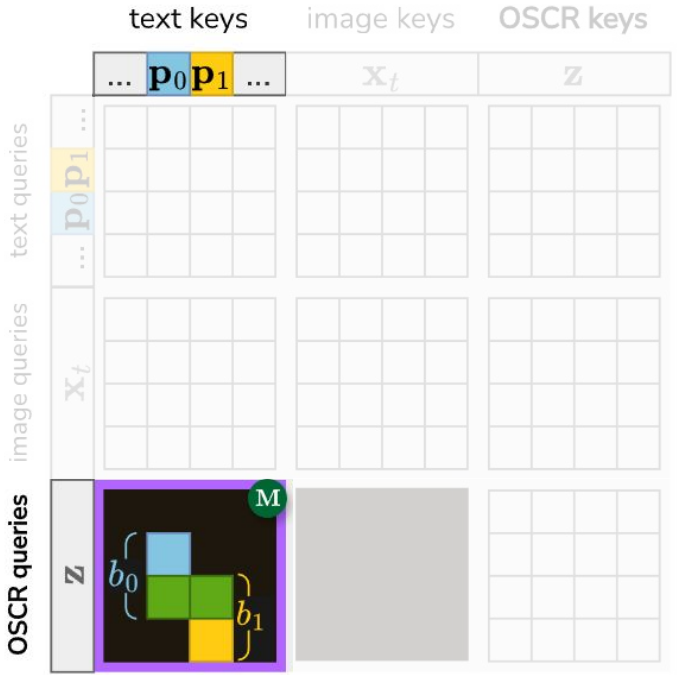
How to bind boxes to corresponding objects?

- **Masked attention!**
- OSCR tokens only attend to their *respective* text tokens, and nothing else.

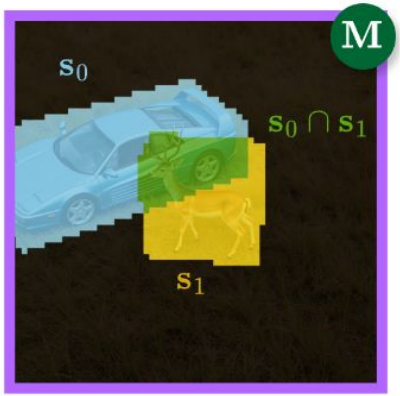


But what about **overlapping** box regions?

(a) Attention inside mmDiT block. Black regions indicate no attention.



(b) Masked attention from OSCR tokens to object tokens in prompt.



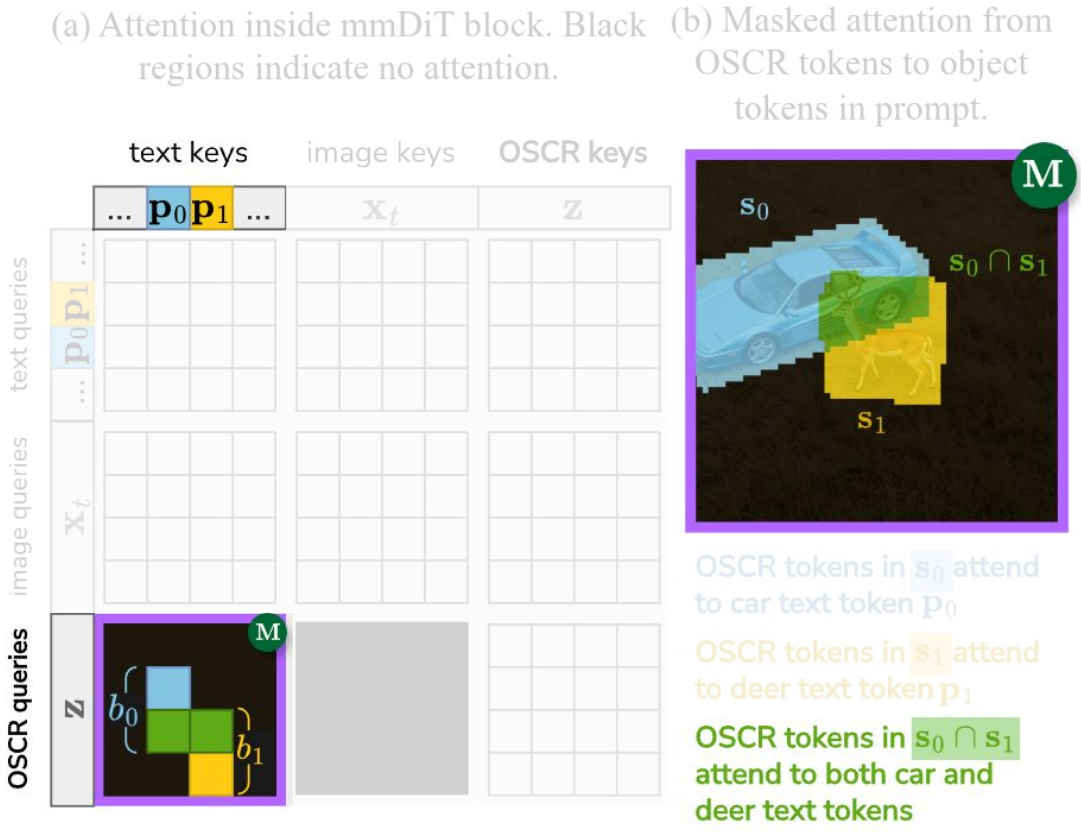
OSCR tokens in s_0 attend to car text token p_0

OSCR tokens in s_1 attend to deer text token p_1

OSCR tokens in $s_0 \cap s_1$ attend to both car and deer text tokens

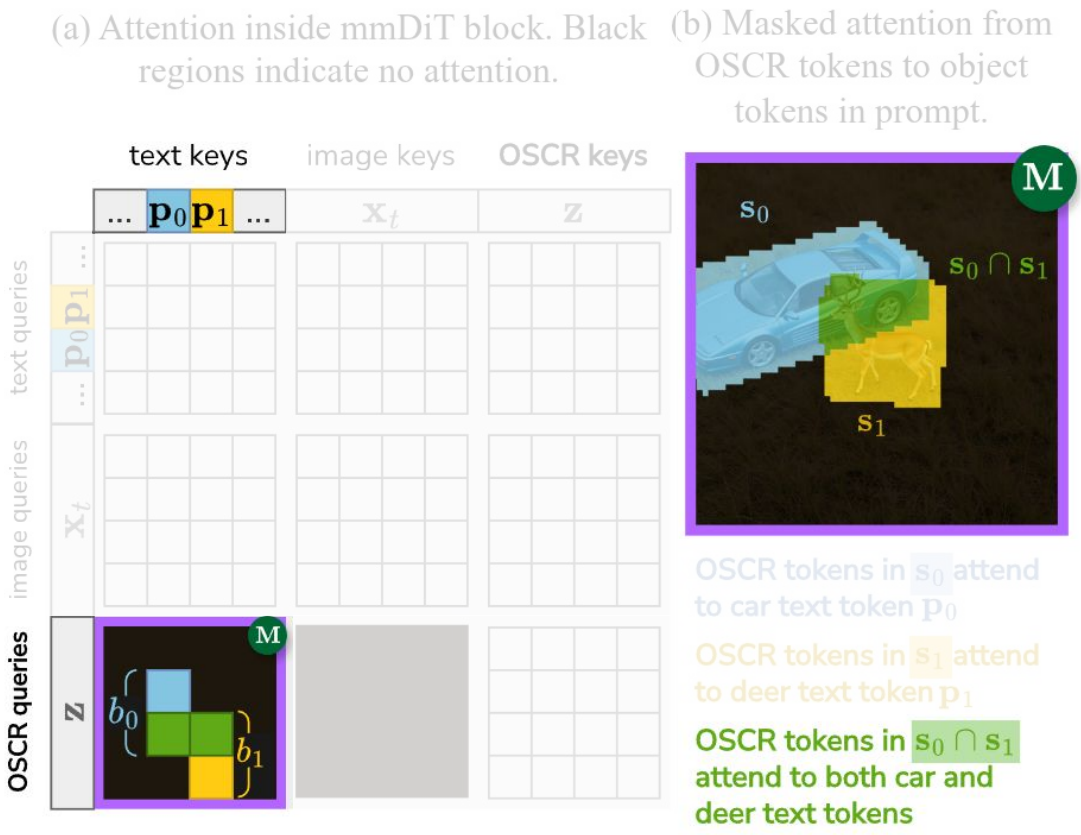
But what about **overlapping** box regions?

- The **green overlap region** attends to multiple objects.



But what about **overlapping** box regions?

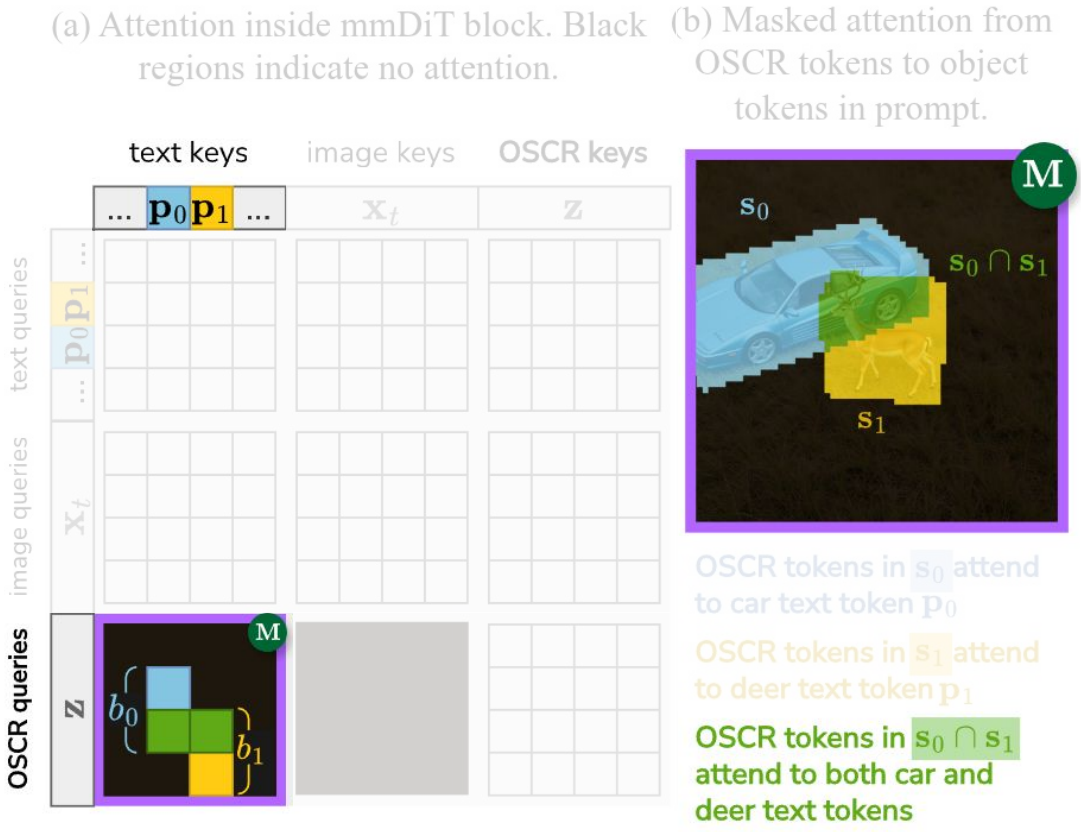
- The **green overlap region** attends to multiple objects.
- **Initial guess:** such attention overlap might lead to attribute mixing between different objects.



But what about **overlapping** box regions?

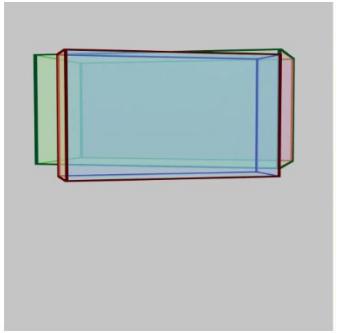
- The **green overlap region** attends to multiple objects.
- **Initial guess:** such attention overlap might lead to attribute mixing between different objects.

But this does not happen!



A closer look.

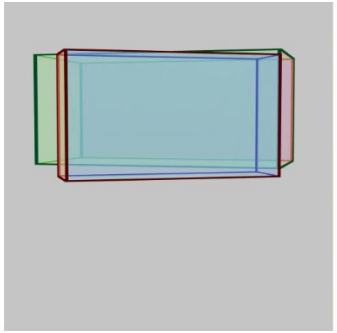
*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*



(a) OSCR condition

A closer look.

*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*



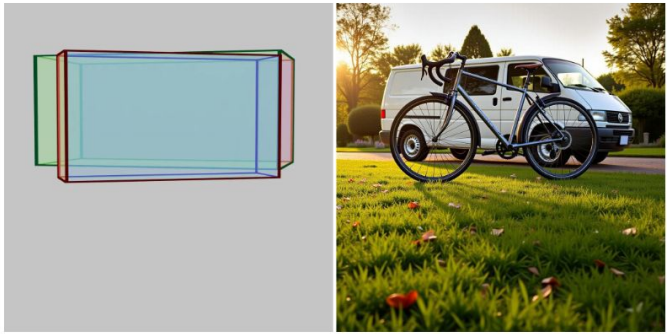
(a) OSCR condition

Hypothesis

Heavy overlap between 3D boxes → overlapping attention → attribute mixing.

A closer look.

*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*



(a) OSCR condition (b) Generated image

Hypothesis	**Observation**
Heavy overlap between 3D boxes → overlapping attention → attribute mixing.	**No attribute mixing**. Sharp occlusion boundaries!

A closer look. **Visualizing cross attention maps.**

*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*

(a) OSCR condition (b) Generated image (c) Bicycle (d) Van

Hypothesis

Heavy overlap between 3D boxes → overlapping attention → attribute mixing.

Observation

No attribute mixing. Sharp occlusion boundaries!

A closer look. **Visualizing cross attention maps.**

*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*

(a) OSCR condition (b) Generated image (c) Bicycle (d) Van

Hypothesis

Heavy overlap between 3D boxes → overlapping attention → attribute mixing.

Observation

No attribute mixing. Sharp occlusion boundaries!

Attention maps directly reveal occlusion boundaries.

A closer look. **Visualizing cross attention maps.**

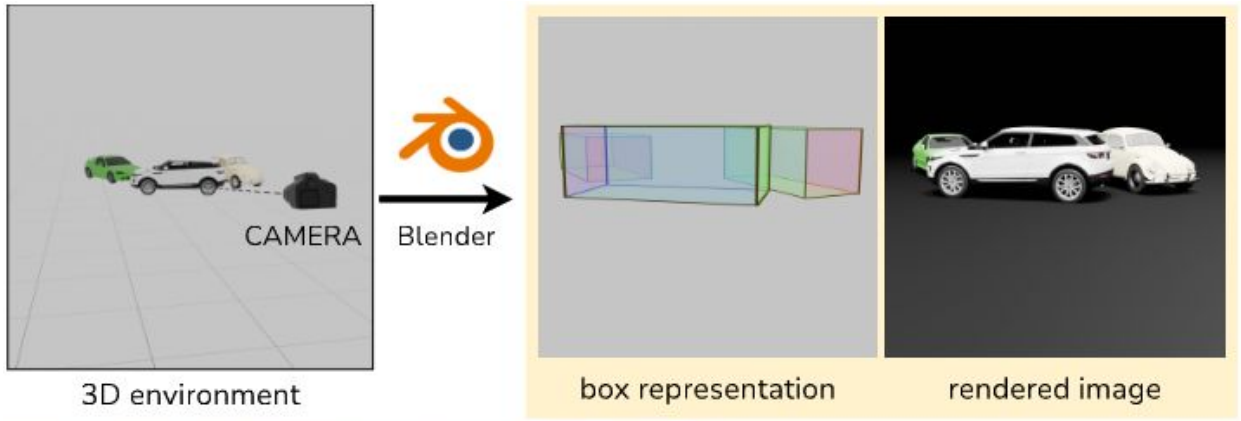
*'A photo of **bicycle** and **van** in a beautiful garden in the morning.'*

(a) OSCR condition (b) Generated image (c) Bicycle (d) Van

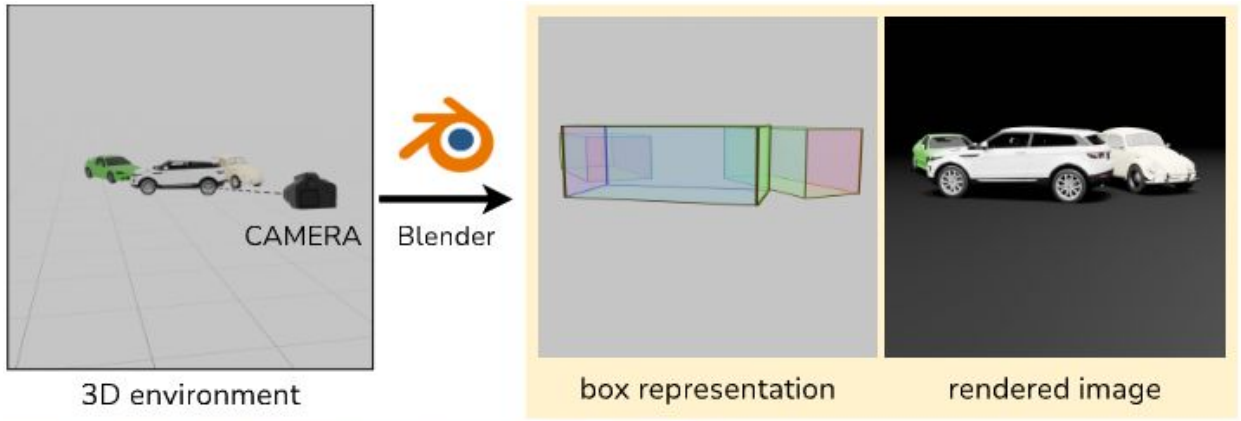
Hypothesis	**Observation**	**Conclusion**
Heavy overlap between 3D boxes → overlapping attention → attribute mixing.	**No attribute mixing**. Sharp occlusion boundaries!	T2I model has **strong priors** for occlusion reasoning.
	Attention maps directly reveal occlusion boundaries.	**SeeThrough3D** effectively leverages these priors.

Training data preparation.

Training data preparation.

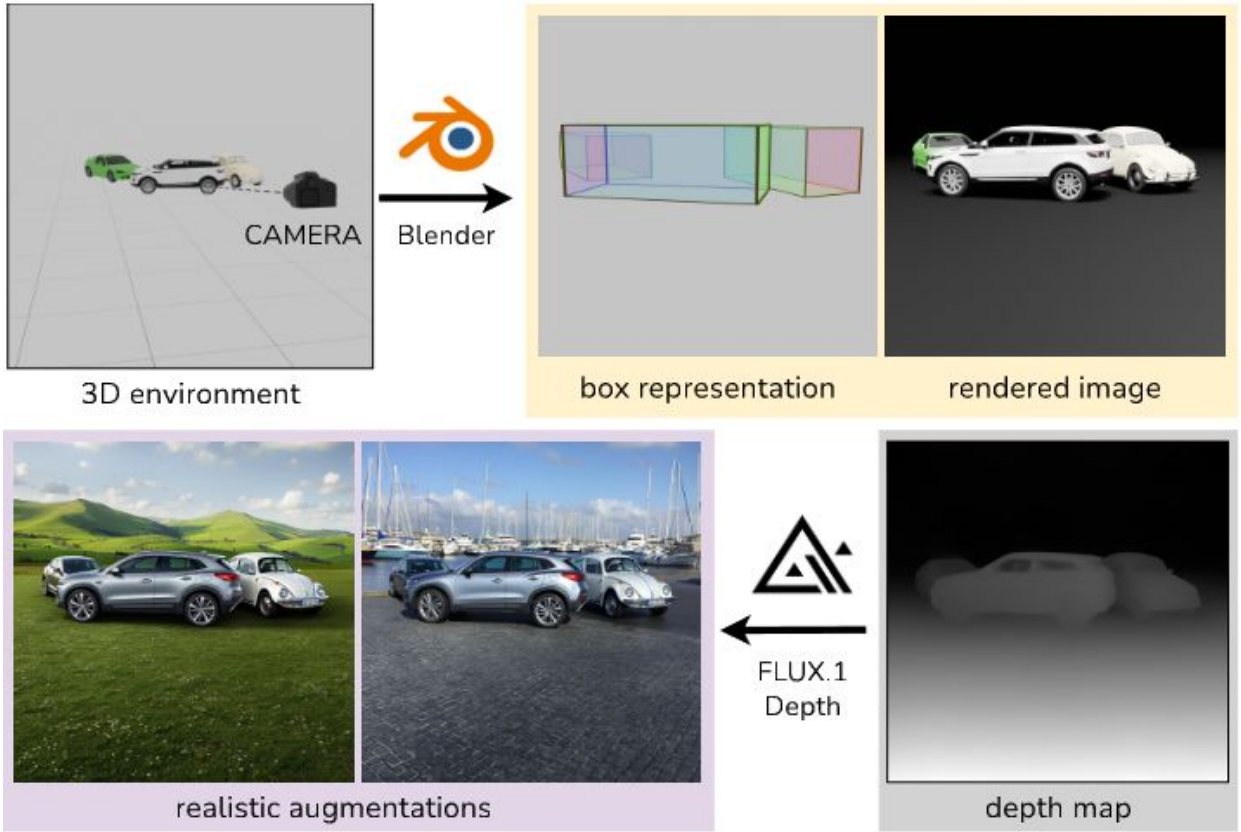


Training data preparation.



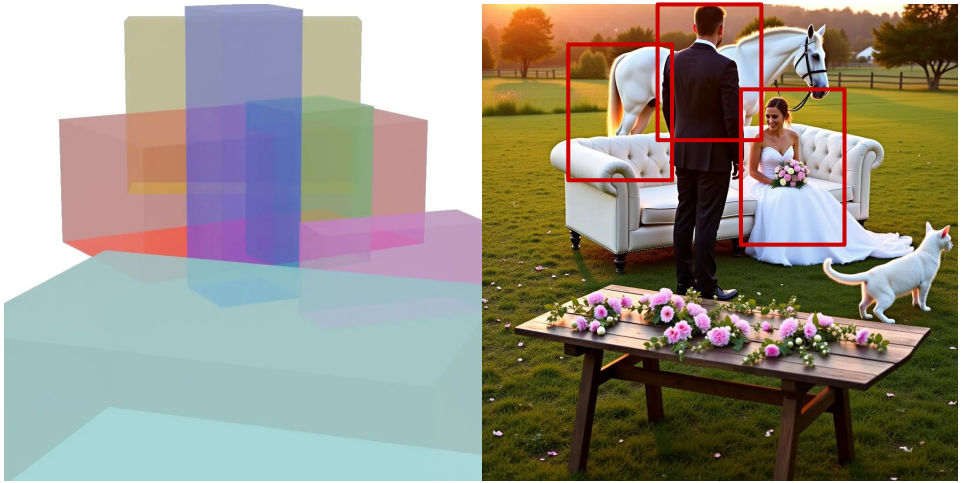
However, training on these rendered images alone can lead to overfitting to synthetic backgrounds!

Training data preparation.



Results.

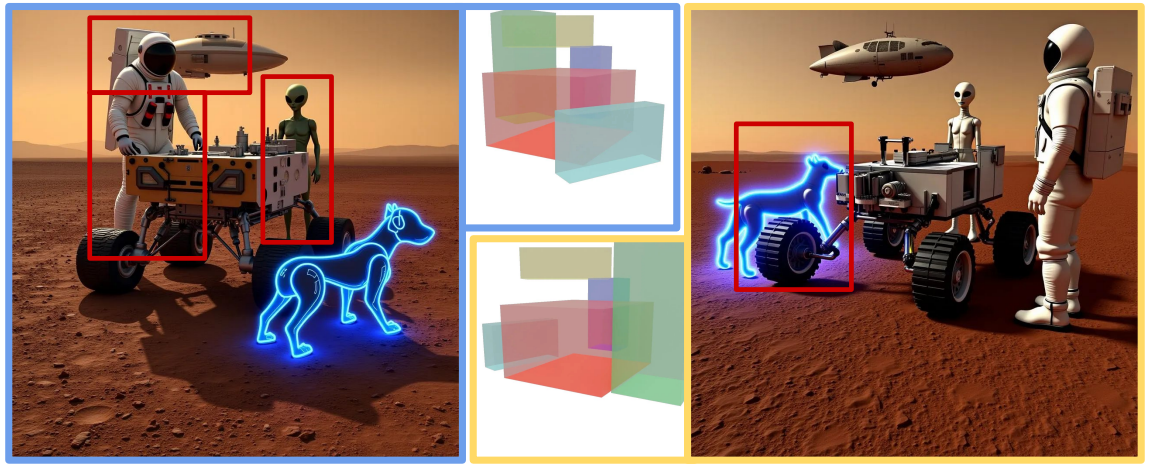
SeeThrough3D adheres to complex layouts.



'A photo of sofa and bride and groom and white horse and table scattered with flowers and cat in a dreamy wedding garden at sunrise.'

Strong **occlusion** consistency.

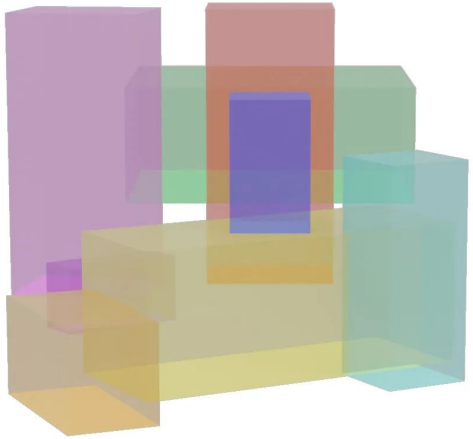
SeeThrough3D adheres to complex layouts.

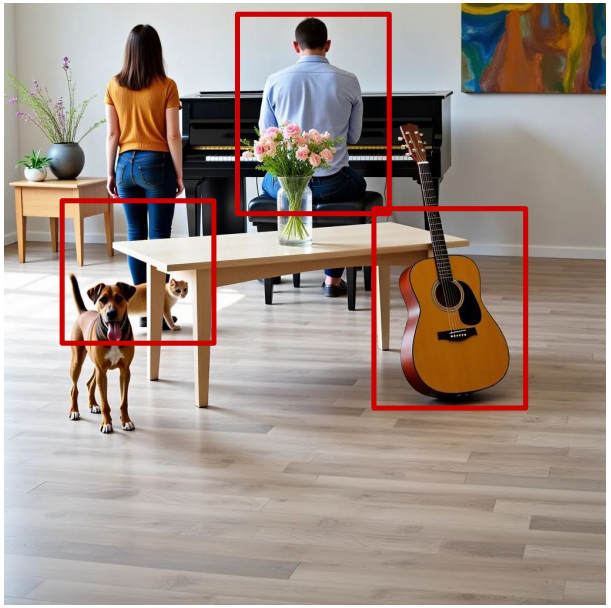


'A photo of mars rover and astronaut and alien and spaceship and blue neon robot dog on Mars, sci-fi scene.'

Strong **occlusion** consistency.

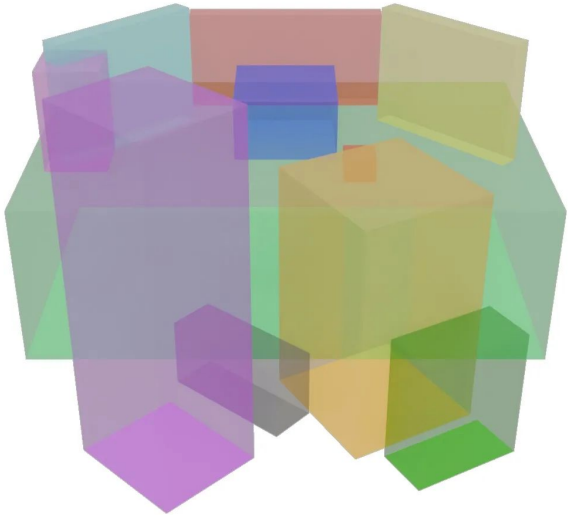
SeeThrough3D adheres to complex layouts.

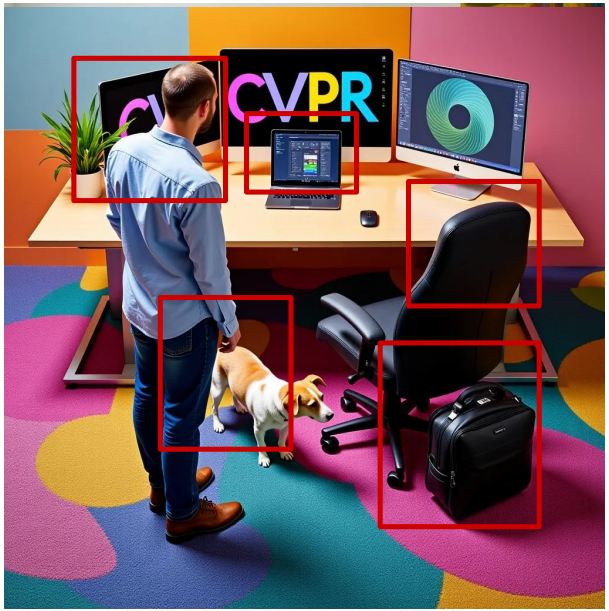




Strong **occlusion** consistency.

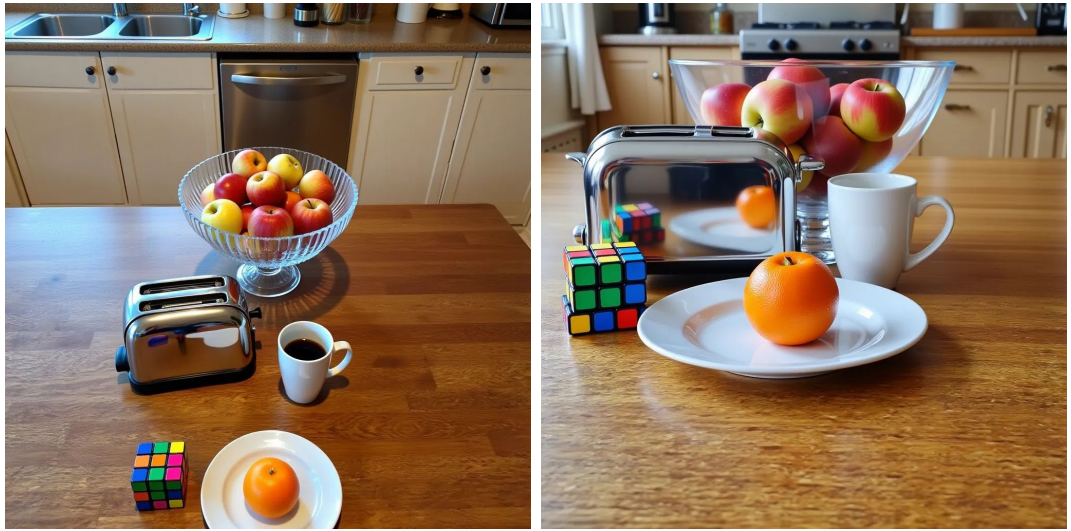
SeeThrough3D adheres to complex layouts.





Strong **occlusion** consistency.

SeeThrough3D enables camera control.



*'A photo of **glass bowl** and **apples** and **shiny metal toaster** and **coffee mug** and **plate** and **orange** and **rubik's cube** on a dining table in a kitchen.'*

SeeThrough3D preserves the T2I prior.

'A photo of red boat with letters 'CVPR' written on it and black boat with 'DENVER' written on it and sedan and man at a beautiful beach at sunset.'

Preserves text rendering capabilities of the T2I model.

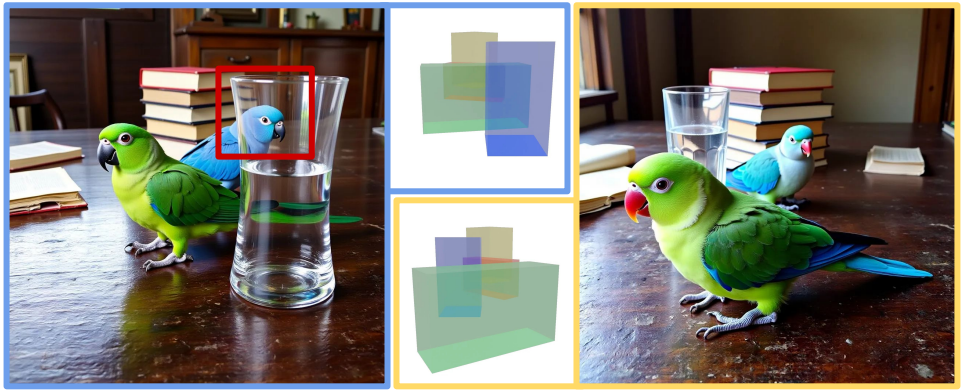
SeeThrough3D preserves the T2I prior.

'A photo of red boat with letters 'CVPR' written on it and black boat with 'DENVER' written on it and sedan and man at a beautiful beach at sunset.'

Preserves text rendering capabilities of the T2I model.

(Text rendering was not seen in our training data)

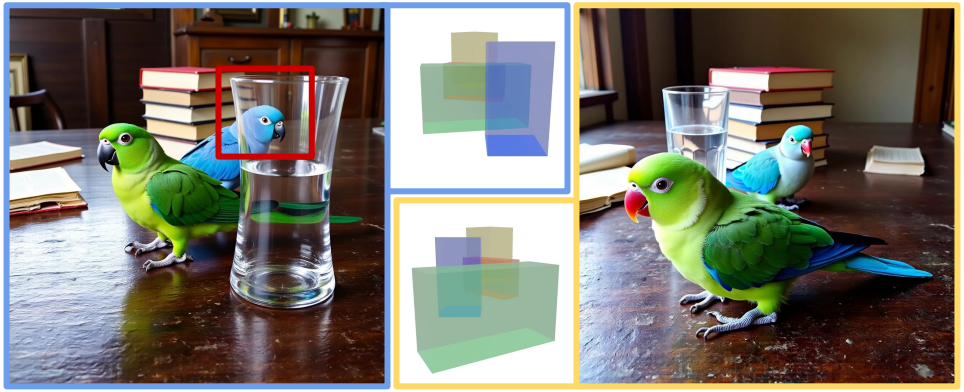
SeeThrough3D preserves the T2I prior.



'A photo of green parrot and blue parrot and transparent glass of water and a stack of books on an old study table.'

Preserves T2I priors for transparency reasoning.

SeeThrough3D preserves the T2I prior.



'A photo of green parrot and blue parrot and transparent glass of water and a stack of books on an old study table.'

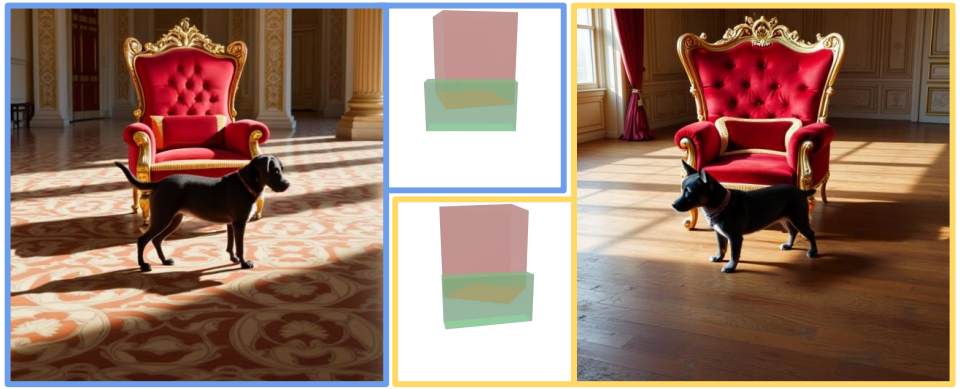
Preserves T2I priors for transparency reasoning.

(Transparent objects or similar lighting phenomena were not seen in our training data)

SeeThrough3D can be personalized!

(a)

<obj0>

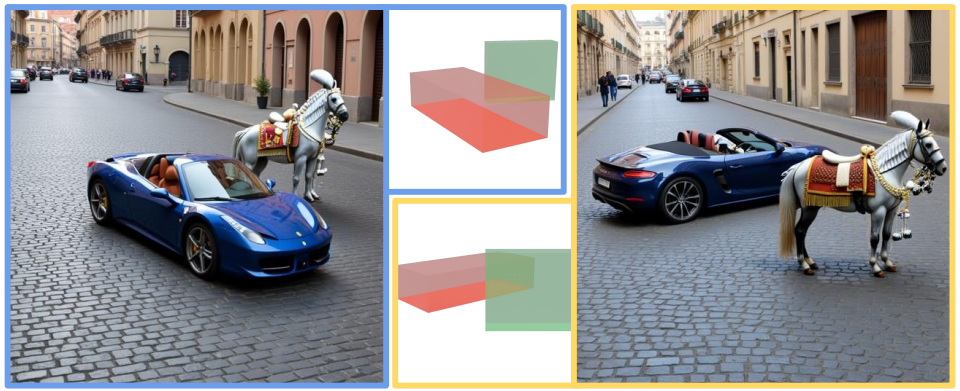


'A photo of <obj0> *and dog in a royal palace.'*

(b)

<obj0>

<obj1>



'A photo of <obj0> *and* <obj1> *on a cobblestone street in a beautiful European city.'*

SeeThrough3D

- First method to enable occlusion aware 3D control in T2I generation.

Project page

SeeThrough3D

- First method to enable occlusion aware 3D control in T2I generation.
- We propose the Occlusion Aware 3D SCene Representation (OSCR), which uses transparency to model hidden regions.

Project page

SeeThrough3D

- First method to enable occlusion aware 3D control in T2I generation.
- We propose the Occlusion Aware 3D SCene Representation (OSCR), which uses transparency to model hidden regions.
- SeeThrough3D effectively leverages the T2I prior to generalize to arbitrary objects.

Project page

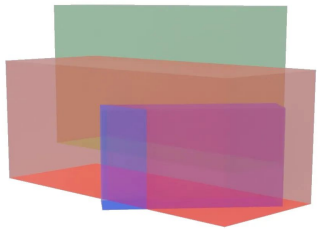
Visit out poster and demo session!

Main poster session: June 6th evening (1645 – 1845 hours).

Demo presentation: June 7th (1045 – 1245 hours) [tentative]

Handling occlusion is key for accurate 3D layout control

- To address this, we propose **SeeThrough3D**, which enables occlusion aware 3D control in T2I generation.

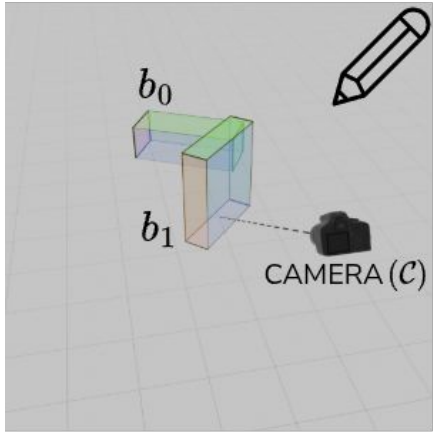


Input layout

Generated image
(**SeeThrough3D**)

From 3D layout to *OSCR*

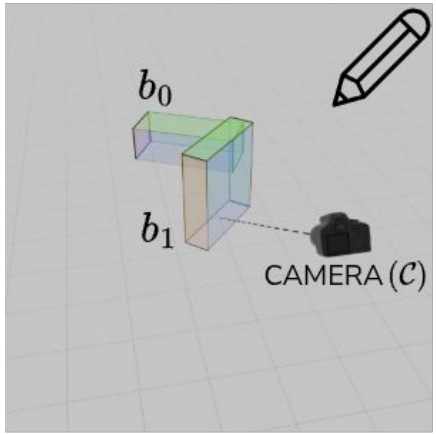
- Objects are arranged as translucent 3D bounding boxes following input layout.



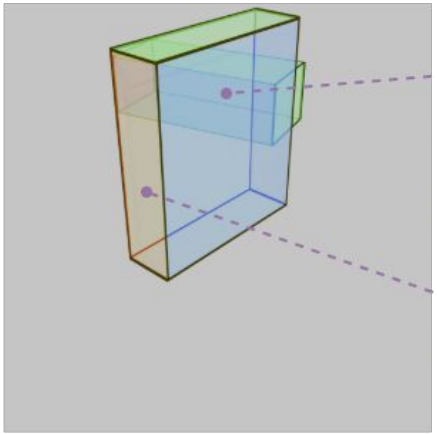
User interface

From 3D layout to *OSCR*

- Objects are arranged as translucent 3D bounding boxes following input layout.
- OSCR representation is rendered through desired camera viewpoint.



User interface



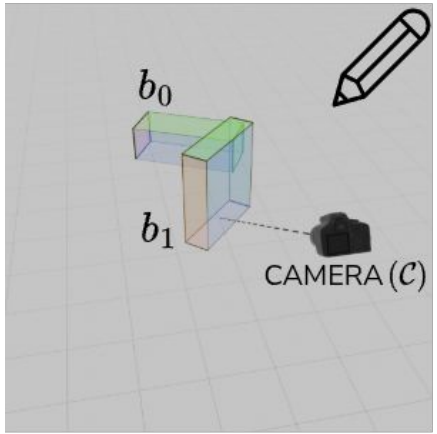
OSCR

Transparency captures occluded objects

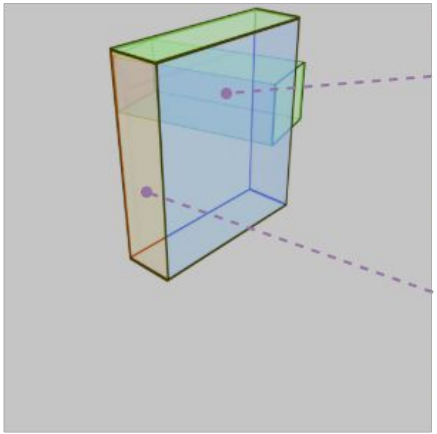
Color-coding faces to indicate 3D orientation (orange for front face, blue for left, others green).

From 3D layout to *OSCR*

- Objects are arranged as translucent 3D bounding boxes following input layout.
- OSCR representation is rendered through desired camera viewpoint.
- Rendered image is used to condition the generative model.



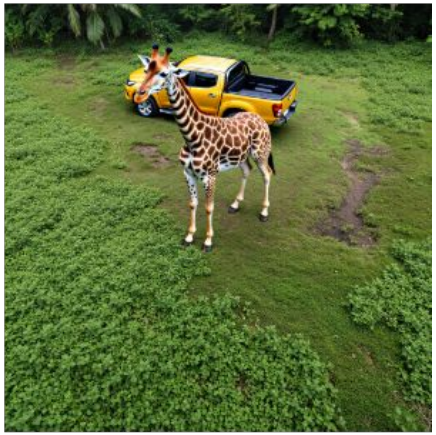
User interface



OSCR

Transparency captures occluded objects

Color-coding faces to indicate 3D orientation (orange for front face, blue for left, others green).



Generated image