

SAM 2: Segment Anything Model 2

SAM 2, the successor to Meta's [Segment Anything Model \(SAM\)](#), is a cutting-edge tool designed for comprehensive object segmentation in both images and videos. It excels in handling complex visual data through a unified, promptable model architecture that supports real-time processing and zero-shot generalization.



Key Features

Ask AI 

How to Run Inference with Meta's SAM2 and MobileSAM Mo...



Watch: How to Run Inference with Meta's SAM2 using Ultralytics | Step-by-Step Guide 🎉

Unified Model Architecture

SAM 2 combines the capabilities of image and video segmentation in a single model. This unification simplifies deployment and allows for consistent performance across different media types. It leverages a flexible prompt-based interface, enabling users to specify objects of interest through various prompt types, such as points, bounding boxes, or masks.

Real-Time Performance

The model achieves real-time inference speeds, processing approximately 44 frames per second. This makes SAM 2 suitable for applications requiring immediate feedback, such as video editing and augmented reality.

Zero-Shot Generalization

SAM 2 can segment objects it has never encountered before, demonstrating strong zero-shot generalization. This is particularly useful in diverse or evolving visual domains where pre-defined categories may not cover all possible objects.

Interactive Refinement

Users can iteratively refine the segmentation results by providing additional prompts, allowing for precise control over the output. This interactivity is essential for fine-tuning results in applications like video annotation or medical imaging.

Advanced Handling of Visual Challenges

SAM 2 includes mechanisms to manage common video segmentation challenges, such as object occlusion and reappearance. It uses a sophisticated memory mechanism to keep track of objects across frames, ensuring continuity even when objects are temporarily obscured or exit and re-enter the scene.

For a deeper understanding of SAM 2's architecture and capabilities, explore the [SAM 2 research paper](#).

Performance and Technical Details

SAM 2 sets a new benchmark in the field, outperforming previous models on various metrics:

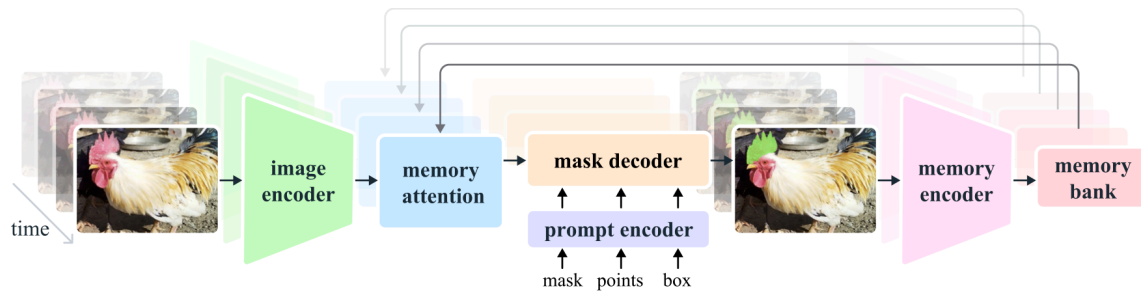
Metric	SAM 2	Previous SOTA
Interactive Video Segmentation	Best	-
Human Interactions Required	3x fewer	Baseline
Image Segmentation Accuracy	Improved	SAM
Inference Speed	6x faster	SAM

Model Architecture

Core Components

- **Image and Video Encoder:** Utilizes a [transformer](#)-based architecture to extract high-level features from both images and video frames. This component is responsible for understanding the visual content at each timestep.
- **Prompt Encoder:** Processes user-provided prompts (points, boxes, masks) to guide the segmentation task. This allows SAM 2 to adapt to user input and target specific objects within a scene.
- **Memory Mechanism:** Includes a memory encoder, memory bank, and memory attention module. These components collectively store and utilize information from past frames, enabling the model to maintain consistent object tracking over time.

- **Mask Decoder:** Generates the final segmentation masks based on the encoded image features and prompts. In video, it also uses memory context to ensure accurate tracking across frames.



Memory Mechanism and Occlusion Handling

The memory mechanism allows SAM 2 to handle temporal dependencies and occlusions in video data. As objects move and interact, SAM 2 records their features in a memory bank. When an object becomes occluded, the model can rely on this memory to predict its position and appearance when it reappears. The occlusion head specifically handles scenarios where objects are not visible, predicting the likelihood of an object being occluded.

Multi-Mask Ambiguity Resolution

In situations with ambiguity (e.g., overlapping objects), SAM 2 can generate multiple mask predictions. This feature is crucial for accurately representing complex scenes where a single mask might not sufficiently describe the scene's nuances.

SA-V Dataset

The SA-V dataset, developed for SAM 2's training, is one of the largest and most diverse video segmentation datasets available. It includes:

- **51,000+ Videos:** Captured across 47 countries, providing a wide range of real-world scenarios.
- **600,000+ Mask Annotations:** Detailed spatio-temporal mask annotations, referred to as "masklets," covering whole objects and parts.
- **Dataset Scale:** It features 4.5 times more videos and 53 times more annotations than previous largest datasets, offering unprecedented diversity and complexity.

Benchmarks

Video Object Segmentation

SAM 2 has demonstrated superior performance across major video segmentation benchmarks:

Dataset	J&F	J	F
DAVIS 2017	82.5	79.8	85.2
YouTube-VOS	81.2	78.9	83.5

Interactive Segmentation

In interactive segmentation tasks, SAM 2 shows significant efficiency and accuracy:

Dataset	NoC@90	AUC
DAVIS Interactive	1.54	0.872

Installation

To install SAM 2, use the following command. All SAM 2 models will automatically download on first use.

```
pip install ultralytics
```

How to Use SAM 2: Versatility in Image and Video Segmentation

The following table details the available SAM 2 models, their pre-trained weights, supported tasks, and compatibility with different operating modes like [Inference](#), [Validation](#), [Training](#), and [Export](#).


Model Type	Pre-trained Weights	Tasks Supported	Inference	Validation	Training
SAM 2 tiny	sam2_t.pt	Instance Segmentation	✓	✗	✗
SAM 2 small	sam2_s.pt	Instance Segmentation	✓	✗	✗
SAM 2 base	sam2_b.pt	Instance Segmentation	✓	✗	✗

Model Type	Pre-trained Weights	Tasks Supported	Inference	Validation	Training
SAM 2 large	sam2_l.pt	Instance Segmentation	✓	✗	✗

SAM 2 Prediction Examples

SAM 2 can be utilized across a broad spectrum of tasks, including real-time video editing, medical imaging, and autonomous systems. Its ability to segment both static and dynamic visual data makes it a versatile tool for researchers and developers.

Segment with Prompts

 **Segment with Prompts**

Use prompts to segment specific objects in images or videos.

Python

```
from ultralytics import SAM

# Load a model
model = SAM("sam2_b.pt")

# Display model information (optional)
model.info()

# Run inference with bboxes prompt
results = model("path/to/image.jpg", bboxes=[100, 100, 200, 200])

# Run inference with single point
results = model(points=[900, 370], labels=[1])

# Run inference with multiple points
results = model(points=[[400, 370], [900, 370]], labels=[1, 1])

# Run inference with multiple points prompt per object
results = model(points=[[400, 370], [900, 370]], labels=[[1, 1]])

# Run inference with negative points prompt
results = model(points=[[400, 370], [900, 370]], labels=[[1, 0]])
```

Segment Everything



Segment Everything

Segment the entire image or video content without specific prompts.

Python

```
from ultralytics import SAM

# Load a model
model = SAM("sam2_b.pt")

# Display model information (optional)
model.info()

# Run inference
model("path/to/video.mp4")
```

CLI

```
# Run inference with a SAM 2 model
yolo predict model=sam2_b.pt source=path/to/video.mp4
```

- This example demonstrates how SAM 2 can be used to segment the entire content of an image or video if no prompts (bboxes/points/masks) are provided.

SAM 2 comparison vs YOLOv8

Here we compare Meta's smallest SAM 2 model, SAM2-t, with Ultralytics smallest segmentation model, [YOLOv8n-seg](#):

Model	Size (MB)	Parameters (M)	Speed (CPU) (ms/im)
Meta SAM-b	375	93.7	161440
Meta SAM2-b	162	80.8	121923
Meta SAM2-t	78.1	38.9	85155
MobileSAM	40.7	10.1	98543
FastSAM-s with YOLOv8 backbone	23.7	11.8	140
Ultralytics YOLOv8n-seg	6.7 (11.7x smaller)	3.4 (11.4x less)	79.5 (1071x faster)

This comparison shows the order-of-magnitude differences in the model sizes and speeds between models. Whereas SAM presents unique capabilities for automatic segmenting, it is not a direct competitor to YOLOv8 segment models, which are smaller, faster and more efficient.

Tests run on a 2023 Apple M2 Macbook with 16GB of RAM using `torch==2.3.1` and `ultralytics==8.3.82`. To reproduce this test:



Example

Python

```
from ultralytics import ASSETS, SAM, YOLO, FastSAM

# Profile SAM2-t, SAM2-b, SAM-b, MobileSAM
for file in ["sam_b.pt", "sam2_b.pt", "sam2_t.pt", "mobile_sam.pt"]:
    model = SAM(file)
    model.info()
    model(ASSETS)

# Profile FastSAM-s
model = FastSAM("FastSAM-s.pt")
model.info()
model(ASSETS)

# Profile YOLOv8n-seg
model = YOLO("yolov8n-seg.pt")
model.info()
model(ASSETS)
```

Auto-Annotation: Efficient Dataset Creation

Auto-annotation is a powerful feature of SAM 2, enabling users to generate segmentation datasets quickly and accurately by leveraging pre-trained models. This capability is particularly useful for creating large, high-quality datasets without extensive manual effort.

How to Auto-Annotate with SAM 2

To auto-annotate your dataset using SAM 2, follow this example:



Auto-Annotation Example

```
from ultralytics.data.annotator import auto_annotate

auto_annotate(data="path/to/images", det_model="yolov8x.pt",
              sam_model="sam2_b.pt")
```

Argument	Type	Description	Default
data	str	Path to a folder containing images to be annotated.	
det_model	str , optional	Pre-trained YOLO detection model. Defaults to 'yolov8x.pt'.	'yolov8x.pt'
sam_model	str , optional	Pre-trained SAM 2 segmentation model. Defaults to 'sam2_b.pt'.	'sam2_b.pt'
device	str , optional	Device to run the models on. Defaults to an empty string (CPU or GPU, if available).	
output_dir	str , None , optional	Directory to save the annotated results. Defaults to a 'labels' folder in the same directory as 'data'.	None

This function facilitates the rapid creation of high-quality segmentation datasets, ideal for researchers and developers aiming to accelerate their projects.

Limitations

Despite its strengths, SAM 2 has certain limitations:

- **Tracking Stability:** SAM 2 may lose track of objects during extended sequences or significant viewpoint changes.
- **Object Confusion:** The model can sometimes confuse similar-looking objects, particularly in crowded scenes.
- **Efficiency with Multiple Objects:** Segmentation efficiency decreases when processing multiple objects simultaneously due to the lack of inter-object communication.
- **Detail Accuracy:** May miss fine details, especially with fast-moving objects. Additional prompts can partially address this issue, but temporal smoothness is not guaranteed.

Citations and Acknowledgements

If SAM 2 is a crucial part of your research or development work, please cite it using the following reference:

BibTeX

```
@article{ravi2024sam2,  
  title={SAM 2: Segment Anything in Images and Videos},  
  author={Ravi, Nikhila and Gabeur, Valentin and Hu, Yuan-Ting and Hu, Ronghang  
and Ryali, Chaitanya and Ma, Tengyu and Khedr, Haitham and R{"a"}dle, Roman and  
Rolland, Chloe and Gustafson, Laura and Mintun, Eric and Pan, Junting and  
Alwala, Kalyan Vasudev and Carion, Nicolas and Wu, Chao-Yuan and Girshick, Ross  
and Doll{"a"}r, Piotr and Feichtenhofer, Christoph},  
  journal={arXiv preprint},  
  year={2024}  
}
```

We extend our gratitude to Meta AI for their contributions to the AI community with this groundbreaking model and dataset.

FAQ

What is SAM 2 and how does it improve upon the original Segment Anything Model (SAM)?

SAM 2, the successor to Meta's [Segment Anything Model \(SAM\)](#), is a cutting-edge tool designed for comprehensive object segmentation in both images and videos. It excels in handling complex visual data through a unified, promptable model architecture that supports real-time processing and zero-shot generalization. SAM 2 offers several improvements over the original SAM, including:

- **Unified Model Architecture:** Combines image and video segmentation capabilities in a single model.
- **Real-Time Performance:** Processes approximately 44 frames per second, making it suitable for applications requiring immediate feedback.
- **Zero-Shot Generalization:** Segments objects it has never encountered before, useful in diverse visual domains.
- **Interactive Refinement:** Allows users to iteratively refine segmentation results by providing additional prompts.
- **Advanced Handling of Visual Challenges:** Manages common video segmentation challenges like object occlusion and reappearance.

For more details on SAM 2's architecture and capabilities, explore the [SAM 2 research paper](#).

How can I use SAM 2 for real-time video segmentation?

SAM 2 can be utilized for real-time video segmentation by leveraging its promptable interface and real-time inference capabilities. Here's a basic example:



Segment with Prompts

Use prompts to segment specific objects in images or videos.

Python

```
from ultralytics import SAM

# Load a model
model = SAM("sam2_b.pt")

# Display model information (optional)
model.info()

# Segment with bounding box prompt
results = model("path/to/image.jpg", bboxes=[100, 100, 200, 200])

# Segment with point prompt
results = model("path/to/image.jpg", points=[150, 150], labels=[1])
```

For more comprehensive usage, refer to the [How to Use SAM 2](#) section.

What datasets are used to train SAM 2, and how do they enhance its performance?

SAM 2 is trained on the SA-V dataset, one of the largest and most diverse video segmentation datasets available. The SA-V dataset includes:

- **51,000+ Videos:** Captured across 47 countries, providing a wide range of real-world scenarios.
- **600,000+ Mask Annotations:** Detailed spatio-temporal mask annotations, referred to as "masklets," covering whole objects and parts.
- **Dataset Scale:** Features 4.5 times more videos and 53 times more annotations than previous largest datasets, offering unprecedented diversity and complexity.

This extensive dataset allows SAM 2 to achieve superior performance across major video segmentation benchmarks and enhances its zero-shot generalization capabilities. For more information, see the [SA-V Dataset](#) section.

How does SAM 2 handle occlusions and object reappearances in video segmentation?

SAM 2 includes a sophisticated memory mechanism to manage temporal dependencies and occlusions in video data. The memory mechanism consists of:

- **Memory Encoder and Memory Bank:** Stores features from past frames.
- **Memory Attention Module:** Utilizes stored information to maintain consistent object tracking over time.
- **Occlusion Head:** Specifically handles scenarios where objects are not visible, predicting the likelihood of an object being occluded.

This mechanism ensures continuity even when objects are temporarily obscured or exit and re-enter the scene. For more details, refer to the [Memory Mechanism and Occlusion Handling](#) section.

How does SAM 2 compare to other segmentation models like YOLOv8?

SAM 2 and Ultralytics YOLOv8 serve different purposes and excel in different areas. While SAM 2 is designed for comprehensive object segmentation with advanced features like zero-shot generalization and real-time performance, YOLOv8 is optimized for speed and efficiency in [object detection](#) and segmentation tasks. Here's a comparison:

Model	Size (MB)	Parameters (M)	Speed (CPU) (ms/im)
Meta SAM-b	375	93.7	161440
Meta SAM2-b	162	80.8	121923
Meta SAM2-t	78.1	38.9	85155
MobileSAM	40.7	10.1	98543
FastSAM-s with YOLOv8 backbone	23.7	11.8	140
Ultralytics YOLOv8n-seg	6.7 (11.7x smaller)	3.4 (11.4x less)	79.5 (1071x faster)

For more details, see the [SAM 2 comparison vs YOLOv8](#) section.



Created 2 months ago



Updated 3 days ago



 Tweet

 Share

-  EN
-  ZH
-  KO
-  JA
-  RU
-  DE
-  FR
-  ES
-  PT
-  IT
-  AR
-  TR
-  VI