Package: geosam (via r-universe)

May 20, 2026

Title Geospatial Image Segmentation with SAM3

Version 0.1.2.9000

Date 2026-05-20

Description Native R API for geospatial image segmentation using Meta's Segment Anything Model 3 (SAM3). Provides tidyverse-friendly interfaces for detecting objects in satellite imagery using text prompts, point prompts, and exemplar-based detection. Features open-vocabulary object detection powered by SAM3's text prompting. Built-in support for downloading satellite imagery from Mapbox, Esri, and MapTiler.

License MIT + file LICENSE

URL https://github.com/walkerke/geosam

BugReports https://github.com/walkerke/geosam/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports reticulate (>= 1.34), sf (>= 1.0), cli (>= 3.0), rlang (>= 1.0), rappdirs, httr2, magick, terra, mapgl, shiny, base64enc

Suggests stars, mapboxapi, RColorBrewer, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

SystemRequirements Python (>= 3.12), SAM3 (https://github.com/facebookresearch/sam3)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libmagick++-dev gsfonts libpng-dev libuv1-dev libssl-dev libproj-dev python3 libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://walkerke.r-universe.dev

Date/Publication 2026-05-20 20:38:15 UTC

RemoteUrl https://github.com/walkerke/geosam

RemoteRef HEAD

RemoteSha 2fa6d48fae691c244da618e716191b81ad35d9f0

Contents

geosam_clear_cache *Clear Imagery Cache*

Description

Clears cached satellite imagery tiles.

Usage

```
geosam_clear_cache()
```

Value

Invisibly returns TRUE.

geosam_configure *Configure geosam Options*

Description

Set package-wide configuration options for model selection and device.

Usage

```
geosam_configure(model = NULL, device = NULL, mapbox_token = NULL)
```

Arguments

model	Model to use: "sam2" (default, latest) or "sam" (original).
device	Computing device: "auto" (default), "mps", "cuda", or "cpu". "auto" selects the best available device.
mapbox_token	Mapbox access token for satellite imagery. Can also be set via the MAPBOX_ACCESS_TOKEN environment variable.

Value

Invisibly returns the previous configuration.

Examples

```
## Not run:
# Use SAM 2 on CPU
geosam_configure(model = "sam2", device = "cpu")

# Set Mapbox token
geosam_configure(mapbox_token = "pk.xxx")

## End(Not run)
```

geosam_diagnose *Diagnose geosam Installation Issues*

Description

Scans the system for conda installations and geosam environments to help troubleshoot installation problems. Useful when you have multiple conda distributions installed (miniconda, miniforge, anaconda, etc.).

Usage

```
geosam_diagnose()
```

Value

A list with diagnostic information, invisibly.

Examples

```
## Not run:
geosam_diagnose()

## End(Not run)
```

geosam_install *Install Python Dependencies for geosam*

Description

Creates a Python virtual environment and installs required packages for SAM3 inference. This only needs to be run once.

Usage

```
geosam_install(
  envname = "geosam",
  method = c("uv", "virtualenv", "conda"),
  gpu = NULL,
  hf_token = NULL,
  python_version = "3.12",
  conda = "auto"
)
```

Arguments

envname	Name of the virtual environment to create (default: "geosam")
method	Installation method: "uv" (recommended, fast), "virtualenv", or "conda". uv is the default and recommended approach.
gpu	Logical. If TRUE, installs GPU-enabled PyTorch. If NULL (default), auto-detects based on available hardware.
hf_token	Optional HuggingFace token for accessing SAM3 model weights. Can also be set via the HF_TOKEN environment variable. Required for SAM3.
python_version	Python version to use (default: "3.12"). SAM3 requires 3.12+.
conda	Path to conda executable. Only used when `method = "conda"`. If "auto" (default), uses reticulate's default conda. Useful when you have multiple conda installations (miniconda, miniforge, anaconda, etc.).

Details

This function installs:

- PyTorch 2.7+ (with MPS support on Apple Silicon, CUDA 12.6+ on NVIDIA GPUs)
- SAM3 from Meta (https://github.com/facebookresearch/sam3)
- rasterio, geopandas, shapely, pyproj (geospatial processing)

Installation Methods:

uv (recommended): Fast, modern Python package manager. Install uv first:

```
# macOS/Linux
curl -LsSf https://astral.sh/uv/install.sh | sh

# Windows (PowerShell)
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"

# Or via pip (any platform)
pip install uv
```

virtualenv: Uses Python's built-in venv via reticulate. Requires Python 3.12+ to be installed on your system.

conda: Uses conda/miniconda. Slower but handles complex dependencies well. If you have multiple conda installations, use the `conda` parameter to specify which one to use.

SAM3 Access:

You must log in to HuggingFace and accept the gated model terms for SAM3 model weights at: https://huggingface.co/facebook/sam3

Value

Invisibly returns TRUE on success.

Examples

```
## Not run:
# Standard installation with uv (recommended)
geosam_install()

# With HuggingFace token
geosam_install(hf_token = "hf_xxxxx")

# Use virtualenv instead of uv
geosam_install(method = "virtualenv")

# Use conda
geosam_install(method = "conda")

# Use a specific conda installation (e.g., miniforge instead of miniconda)
geosam_install(method = "conda", conda = "~/miniforge3/bin/conda")

## End(Not run)
```

`geosam_status` *Check geosam Installation Status*

Description

Checks the Python environment and reports which features are available.

Usage

```
geosam_status(check_access = TRUE)
```

Arguments

`check_access`	Logical. If TRUE, checks whether the configured HuggingFace token can access the gated `facebook/sam3` model metadata. This does not download model weights.

Value

A list with installation status information, invisibly.

Examples

```
## Not run:
geosam_status()

## End(Not run)
```

`get_imagery` *Download Satellite Imagery*

Description

Downloads satellite imagery tiles for a bounding box and saves as a GeoTIFF. Uses R-native tools (httr2, terra) for reliable tile fetching and georeferencing.

Usage

```
get_imagery(
  bbox,
  output = NULL,
  source = c("mapbox", "esri", "maptiler"),
  zoom = 17,
  api_key = NULL
)
```

Arguments

`bbox`	Bounding box for the area. Can be a numeric vector `c(xmin, ymin, xmax, ymax)` in WGS84, or an sf/sfc object.
`output`	Path for output GeoTIFF. If NULL, creates a temp file.
`source`	Imagery source: "mapbox", "esri", or "maptiler". • "mapbox": Requires `MAPBOX_PUBLIC_TOKEN` environment variable • "esri": Free Esri World Imagery (no API key required) • "maptiler": Requires `MAPTILER_API_KEY` environment variable
`zoom`	Tile zoom level (15-19). Higher values = more detail. Recommended: 17-18 for objects like buildings, swimming pools.
`api_key`	API key for the imagery source. For Mapbox, uses MAPBOX_PUBLIC_TOKEN environment variable by default. For MapTiler, uses MAPTILER_API_KEY environment variable by default.

Value

Path to the downloaded GeoTIFF file.

Examples

```
## Not run:
# Download imagery for an area
img <- get_imagery(
  bbox = c(-102.5, 31.8, -102.4, 31.9),
  source = "mapbox",
  zoom = 17
)
```

```
# Use with detection
pads <- sam_detect(image = img, text = "swimming pool")

## End(Not run)
```

`is_geosam` *Check if Object is a geosam*

Description

Check if Object is a geosam

Usage

```
is_geosam(x)
```

Arguments

`x`	Object to test.

Value

Logical indicating if `x` is a geosam object.

`is_geosam_image` *Check if Object is a geosam_image*

Description

Check if Object is a geosam_image

Usage

```
is_geosam_image(x)
```

Arguments

`x`	Object to test.

Value

Logical indicating if `x` is a geosam_image object.

plot.geosam *Plot geosam Detection Results*

Description

Displays satellite imagery with detection polygons overlaid using terra.

Usage

```
## S3 method for class 'geosam'
plot(
  x,
  fill = "#FACC15",
  border = NULL,
  palette = NULL,
  opacity = 0.5,
  lwd = 1,
  main = NULL,
  add = FALSE,
  ...
)
```

Arguments

x	A geosam object from sam_detect() or sam_explore().
fill	Fill color for detection polygons. Default is semi-transparent yellow. Ignored if palette is specified.
border	Border color for polygon outlines. If NULL (default), automatically darkens the fill color. Set to NA to disable borders.
palette	Optional color palette for distinct colors per detection. Can be: • A character vector of colors (recycled if needed) • A palette name from RColorBrewer (e.g., "Set1", "Dark2", "Paired") • A palette function that takes n and returns colors
opacity	Opacity for fill colors (0-1). Default is 0.5. Applied to all fills.
lwd	Line width for polygon borders. Default is 1.
main	Plot title. Default is NULL (no title).
add	If TRUE, add to existing plot instead of creating new one. Default is FALSE.
...	Additional arguments passed to terra::plotRGB() when add = FALSE.

Value

Invisibly returns the sf polygons that were plotted.

Examples

```
## Not run:
result <- sam_detect(
  bbox = c(-102.63, 31.81, -102.62, 31.83),
  text = "swimming pool"
)
plot(result)

# Distinct colors per detection
plot(result, palette = "Set1")

# Custom colors
plot(result, fill = "red")  # border auto-darkens

# Layer multiple results
plot(result1)
plot(result2, add = TRUE, fill = "blue")

## End(Not run)
```

`plot.geosam_image` *Plot geosam_image Detection Results*

Description

Displays an image with detection masks overlaid using magick.

Usage

```
## S3 method for class 'geosam_image'
plot(
  x,
  fill = "#FACC15",
  border = NULL,
  palette = NULL,
  opacity = 0.4,
  border_width = 2,
  add = FALSE,
  base_img = NULL,
  ...
)
```

Arguments

`x`	A geosam_image object from `sam_image()` or `sam_explore_image()`.
`fill`	Fill color for detection masks. Default is yellow. Ignored if `palette` is specified.
`border`	Border color for detection outlines. If NULL (default), automatically darkens the fill color. Set to NA to disable borders.

palette	Optional color palette for distinct colors per detection. Can be: • A character vector of colors (recycled if needed) • A palette name from RColorBrewer (e.g., "Set1", "Dark2", "Paired") • A palette function that takes n and returns colors
opacity	Opacity of the mask overlay (0-1). Default is 0.4.
border_width	Width of border lines in pixels. Default is 2.
add	If TRUE, add to an existing magick image instead of loading from file. Pass the image as the x parameter's image_path will be ignored and base_img must be provided.
base_img	A magick image to add overlays to. Required when add = TRUE.
...	Additional arguments (currently ignored).

Value

Invisibly returns the composite magick image, which can be saved with `magick::image_write()` or further manipulated.

Examples

```
## Not run:
result <- sam_image("photo.jpg", text = "dog")
plot(result)

# Distinct colors per detection
plot(result, palette = "Set2")

# Custom color (border auto-matches)
plot(result, fill = "red")

# Layer multiple detections
img <- plot(result1)
img <- plot(result2, add = TRUE, base_img = img, fill = "blue")
img <- plot(result3, add = TRUE, base_img = img, fill = "red")

# Save the result
magick::image_write(img, "annotated.png")

## End(Not run)
```

---

print.geosam *Print Method for geosam Objects*

Description

Print Method for geosam Objects

Usage

```
## S3 method for class 'geosam'
print(x, ...)
```

Arguments

x	A geosam object.
...	Additional arguments (ignored).

Value

Invisibly returns x.

---

print.geosam_image *Print Method for geosam_image Objects*

Description

Print Method for geosam_image Objects

Usage

```
## S3 method for class 'geosam_image'
print(x, ...)
```

Arguments

x	A geosam_image object.
...	Additional arguments (ignored).

Value

Invisibly returns x.

sam_as_matrix *Extract Masks as Matrices*

Description

Returns detection masks as a list of R matrices.

Usage

```
sam_as_matrix(x)
```

Arguments

x	A geosam or geosam_image object.

Value

A list of binary matrices (1 = detected, 0 = background).

Examples

```
## Not run:
result <- sam_image("photo.jpg", text = "dog")
masks <- sam_as_matrix(result)
image(masks[[1]])  # Display first mask

## End(Not run)
```

sam_as_raster *Extract Raster Masks from geosam Object*

Description

Converts detection masks from a geosam object to a terra SpatRaster.

Usage

```
sam_as_raster(x, combined = FALSE)
```

Arguments

x	A geosam object from `sam_detect()` or `sam_explore()`.
combined	If TRUE, combine all masks into a single layer (union). If FALSE (default), return a multi-layer raster with one layer per detection.

Value

A terra SpatRaster with mask values (1 = detected, 0 = background). Returns NULL if no masks.

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "building")
mask_raster <- sam_as_raster(result, combined = TRUE)
terra::plot(mask_raster)

## End(Not run)
```

---

sam_as_sf *Extract sf Polygons from geosam or geosam_image Object*

Description

Converts detection masks to sf polygon geometries.

Usage

```
sam_as_sf(x, min_area = NULL, max_area = NULL)
```

Arguments

x	A geosam or geosam_image object.
min_area	Minimum object area. For geosam, in square meters. For geosam_image, in square pixels. Objects smaller are filtered out.
max_area	Maximum object area. For geosam, in square meters. For geosam_image, in square pixels. Objects larger are filtered out.

Details

For geosam objects (georeferenced imagery), returns polygons in WGS84 coordinates with area in square meters.

For geosam_image objects (non-georeferenced images), returns polygons in pixel coordinates where x = column (from left) and y = row (from top).

Value

An sf data frame with polygon geometries, scores, and area. Returns NULL if no polygons remain after filtering.

Examples

```
## Not run:
# Georeferenced imagery
result <- sam_detect(image = "satellite.tif", text = "building")
buildings <- sam_as_sf(result, min_area = 100)

# Non-georeferenced image
result <- sam_image("photo.jpg", text = "dog")
dogs <- sam_as_sf(result)  # coordinates in pixels

## End(Not run)
```

`sam_batch` *Batch Process Multiple Areas*

Description

Runs SAM3 detection on multiple areas (polygons) and returns combined results.

Usage

```
sam_batch(
  areas,
  text,
  source = "mapbox",
  zoom = 17,
  threshold = 0.5,
  min_area = NULL,
  max_area = NULL,
  .progress = TRUE
)
```

Arguments

`areas`	An sf object with polygon geometries defining areas to process.
`text`	Text prompt for detection.
`source`	Imagery source: "mapbox", "esri", or "maptiler".
`zoom`	Tile zoom level for imagery download.
`threshold`	Detection confidence threshold.
`min_area`	Minimum object area in square meters.
`max_area`	Maximum object area in square meters.
`.progress`	Logical. If TRUE, shows a progress bar.

Value

An sf data frame with detection results. Includes an `area_id` column linking detections to input areas.

Examples

```
## Not run:
library(tigris)

# Process multiple counties
counties <- counties("TX") |>
  filter(NAME %in% c("Midland", "Ector"))

pads <- sam_batch(
  areas = counties,
  text = "swimming pool",
  min_area = 500
)

## End(Not run)
```

---

sam_bbox *Get Bounding Box of Detections*

Description

Returns the bounding box encompassing all detected objects.

Usage

```
sam_bbox(x)
```

Arguments

x	A geosam object from `sam_detect()` or `sam_explore()`.

Value

An sf bbox object, or NULL if no detections.

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "building")
bbox <- sam_bbox(result)

## End(Not run)
```

sam_count *Get Number of Detections*

Description

Returns the number of detected objects.

Usage

```
sam_count(x)
```

Arguments

x	A geosam or geosam_image object.

Value

Integer count of detections.

sam_detect *Detect Objects in Geospatial Imagery Using SAM3*

Description

Main function for object detection using Meta's SAM3 model. Supports text prompts, point prompts, box prompts, and exemplar-based detection.

Usage

```
sam_detect(
  image = NULL,
  bbox = NULL,
  text = NULL,
  boxes = NULL,
  points = NULL,
  labels = NULL,
  exemplar = NULL,
  source = "mapbox",
  zoom = 17,
  threshold = 0.5,
  chunked = NULL,
  chunk_size = NULL,
  chunk_overlap = NULL,
  min_area = NULL,
  max_area = NULL
)
```

Arguments

image	Path to a GeoTIFF image, or NULL to download imagery for bbox. Large images (>2000 pixels on longest dimension) are automatically processed in chunks when using text prompts.
bbox	Bounding box for the area of interest. Can be a numeric vector `c(xmin, ymin, xmax, ymax)` in WGS84, or an sf/sfc object.
text	Text prompt describing objects to detect (e.g., "swimming pool", "swimming pool", "solar panel"). Uses SAM3's open-vocabulary detection.
boxes	sf object with polygons/boxes to use as box prompts.
points	sf object with points to use as point prompts.
labels	Integer vector of labels for point prompts (1 = foreground, 0 = background). If NULL, all points are treated as foreground.
exemplar	sf polygon representing an example object. SAM3 will find all similar objects in the image.
source	Imagery source if downloading: "mapbox", "esri", or "maptiler".
zoom	Tile zoom level for imagery download (17-19 recommended).
threshold	Detection confidence threshold (0-1). Lower values return more detections.
chunked	Control chunking for large areas/images: • `NULL` (default): Auto-chunk when image >2000px or bbox requires multiple tiles • `TRUE`: Force chunked detection • `FALSE`: Disable chunking (may cause memory issues for large images)
chunk_size	Target chunk size in pixels when chunking. Defaults to an internal value tuned for SAM3.
chunk_overlap	Overlap in pixels when chunking. Defaults to an internal value that reduces tile-boundary artifacts.
min_area	Minimum object area in square meters. Objects smaller than this are filtered out. For chunked detection, filtering happens during processing (better performance). Default NULL (no minimum).
max_area	Maximum object area in square meters. Objects larger than this are filtered out. Default NULL (no maximum).

Details

For large areas or images, detection is automatically chunked to maintain accuracy and avoid memory issues. This means you can pass a large bounding box (e.g., a census tract) or a large GeoTIFF file (e.g., an orthoimage) and get reliable detection without running out of memory.

Value

A `geosam` object containing detection masks and metadata. Use `sam_as_sf()` to extract polygons, `sam_filter()` to filter by area/score. Returns NULL if no objects are detected.

Examples

```
## Not run:
# Text-based detection
result <- sam_detect(
  bbox = c(-102.5, 31.8, -102.4, 31.9),
  text = "swimming pool"
)
pads <- result |> sam_filter(min_area = 500) |> sam_as_sf()

# Large area - automatically chunked
result <- sam_detect(
  bbox = c(-118.45, 34.08, -118.40, 34.12),  # ~5km area
  text = "swimming pool",
  zoom = 18
)

# Large user-provided image - automatically chunked
result <- sam_detect(
  image = "large_orthoimage.tif",
  text = "trees",
  threshold = 0.3
)

# Point prompts on existing image
result <- sam_detect(
  image = "satellite.tif",
  points = my_points_sf
)

# Box prompts
result <- sam_detect(
  image = "satellite.tif",
  boxes = my_boxes_sf
)

# Exemplar-based detection
result <- sam_detect(
  image = "satellite.tif",
  exemplar = my_example_polygon
)

## End(Not run)
```

---

sam_explore *Interactive Discovery and Detection*

Description

Opens an interactive map to explore satellite imagery, draw prompts, and run SAM detection.

Usage

```
sam_explore(
  source = c("mapbox", "esri", "maptiler"),
  center = NULL,
  bbox = NULL,
  zoom = 15,
  units = c("metric", "imperial"),
  quality = c("balanced", "fast", "accurate"),
  ...
)
```

Arguments

source	Imagery source: "mapbox", "esri", or "maptiler". • "mapbox": Requires MAPBOX_PUBLIC_TOKEN environment variable • "esri": Free Esri World Imagery (no API key required) • "maptiler": Requires MAPTILER_API_KEY environment variable
center	Initial map center as c(lng, lat). If NULL, defaults to US center.
bbox	Initial bounding box as c(xmin, ymin, xmax, ymax) or sf object. If provided, map will zoom to this extent.
zoom	Initial zoom level (default 15).
units	Unit system for the scale bar and area display: "metric" (km/m) or "imperial" (mi/ft). Default is "metric".
quality	Initial detection quality preset: "balanced" (default, chunked with moderate overlap), "fast" (single-pass, no chunking — best for small areas), or "accurate" (chunked with smaller tiles and more overlap). Exposed in the UI under "Advanced".
...	Additional arguments passed to sam_detect(). Useful for setting min_area, max_area, or other detection parameters.

Details

The explorer provides a complete workflow:

1. Navigate the map to find an area of interest
2. Select a prompt type (text, box, points, or exemplar)
3. Enter text or draw prompts on the map
4. Click "Detect" to run SAM on the current viewport
5. View results and optionally refine with +/- points
6. Click "Done" to return the geosam object

Value

A geosam object when the user clicks "Done", or NULL if cancelled.

Examples

```
## Not run:
# Start exploring with Mapbox satellite
result <- sam_explore(source = "mapbox", center = c(-102.5, 31.8), zoom = 15)

# Extract results
if (!is.null(result)) {
  polygons <- sam_as_sf(result)
}

## End(Not run)
```

`sam_explore_image` *Interactive Image Exploration and Detection*

Description

Opens an interactive viewer to explore an image and run SAM detection using text prompts, point clicks, or drawn boxes.

Usage

```
sam_explore_image(image)
```

Arguments

`image`	Path to image file (PNG, JPG, etc.). Required.

Details

The explorer provides a complete workflow:

1. View and pan/zoom the image
2. Select a prompt type (text, box, or points)
3. Enter text or draw/click prompts on the image
4. Click "Detect" to run SAM
5. View results
6. Click "Done" to return the geosam_image object

The viewer uses MapLibre with a blank background and the image displayed as a raster layer, enabling all standard map interactions.

Value

A `geosam_image` object when the user clicks "Done", or NULL if cancelled.

Examples

```
## Not run:
# Explore a photo
result <- sam_explore_image("photo.jpg")

# Extract detected regions
if (!is.null(result)) {
  polygons <- sam_as_sf(result)
}

## End(Not run)
```

`sam_filter` *Filter Detections by Area or Score*

Description

Filters detections in a geosam object based on area and/or confidence score.

Usage

```
sam_filter(x, min_area = NULL, max_area = NULL, min_score = NULL)
```

Arguments

`x`	A geosam object.
`min_area`	Minimum area in square meters.
`max_area`	Maximum area in square meters.
`min_score`	Minimum confidence score (0-1).

Value

A new geosam object with filtered detections.

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "building")
filtered <- result |> sam_filter(min_area = 100, min_score = 0.7)

## End(Not run)
```

sam_find_similar *Find Similar Objects Using Selected Detection as Exemplar*

Description

Uses a single selected detection as an exemplar to find all similar objects in the image.

Usage

```
sam_find_similar(x)
```

Arguments

x	A geosam object with exactly one detection (use `sam_select()` first).

Value

A new geosam object with all detected similar objects.

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "swimming pool")
# View results, pick the best one, find all similar
similar <- result |>
  sam_select(3) |>
  sam_find_similar()

## End(Not run)
```

sam_image *Detect Objects in a Local Image*

Description

Run SAM3 object detection on a non-georeferenced image (PNG, JPG, etc.). For georeferenced satellite imagery, use `sam_detect()` instead.

Usage

```
sam_image(
  image,
  text = NULL,
  points = NULL,
  labels = NULL,
  boxes = NULL,
  exemplar = NULL,
  threshold = 0.5
)
```

Arguments

image	Path to image file (PNG, JPG, TIFF, etc.).
text	Text prompt describing objects to detect (e.g., "car", "person").
points	Matrix or data.frame of point coordinates in pixels (x, y). Column 1 is x (horizontal), column 2 is y (vertical, from top).
labels	Integer vector of labels for point prompts (1 = foreground, 0 = background). If NULL, all points are treated as foreground.
boxes	Matrix or data.frame of box coordinates in pixels. Each row: (xmin, ymin, xmax, ymax).
exemplar	Vector of box coordinates for an exemplar (xmin, ymin, xmax, ymax).
threshold	Detection confidence threshold (0-1).

Value

A geosam_image object containing detection masks and metadata. Use sam_as_sf() to extract polygons (in pixel coordinates). Returns NULL if no objects are detected.

Examples

```
## Not run:
# Text-based detection on a photo
result <- sam_image("photo.jpg", text = "dog")

# Point prompts (click locations in pixels)
result <- sam_image("photo.jpg", points = matrix(c(100, 200, 150, 250), ncol = 2, byrow = TRUE))

# Extract polygons
polys <- sam_as_sf(result)

## End(Not run)
```

---

sam_is_loaded *Check if Model is Loaded*

Description

Checks whether the SAM3 model is currently loaded in memory.

Usage

```
sam_is_loaded()
```

Value

Logical indicating whether the model is loaded.

sam_load *Load SAM3 Model*

Description

Loads the SAM3 model into memory. Useful for interactive use or Shiny apps where you want to keep the model warm between calls.

Usage

```
sam_load(device = "auto")
```

Arguments

device	Device to load model on: "auto", "mps", "cuda", or "cpu".

Value

Invisibly returns TRUE on success.

Examples

```
## Not run:
# Load model before making multiple detections
sam_load()

# Now detections are faster
result1 <- sam_detect(image = "img1.tif", text = "building")
result2 <- sam_detect(image = "img2.tif", text = "building")

# Unload when done
sam_unload()

## End(Not run)
```

sam_merge_edges *Merge Polygons Split at Tile Boundaries*

Description

When detecting objects over large areas, geosam uses tiled processing which can split objects that span tile boundaries into multiple polygons. This function merges polygons that are close together, healing those splits.

Usage

```
sam_merge_edges(x, buffer = 2, by_prompt = TRUE)
```

Arguments

x	A geosam object or sf object with detection results.
buffer	Distance in meters to buffer polygons before checking for overlap. Larger values merge polygons that are further apart. Default is 2.
by_prompt	If TRUE and a prompt column exists, only merge polygons with the same prompt value. Default is TRUE.

Details

This function is useful when you notice objects being split at regular intervals (tile boundaries). The default buffer of 2 meters catches most boundary splits without merging truly separate objects.

For aggressive merging of nearby objects (not just boundary splits), use a larger buffer value, but be aware this may merge objects that should remain separate.

Value

An sf object with merged polygons. Scores are aggregated by taking the maximum score from merged polygons.

Examples

```
## Not run:
# Detect buildings over a large area (uses tiling internally)
buildings <- sam_detect(
  bbox = c(-118.5, 34.0, -118.4, 34.1),
  text = "building",
  zoom = 18
)

# Merge any buildings split at tile boundaries
merged <- sam_merge_edges(buildings)

# More aggressive merging (5m buffer)
merged <- sam_merge_edges(buildings, buffer = 5)

## End(Not run)
```

---

sam_scores *Get Detection Scores*

Description

Returns confidence scores for all detections.

Usage

```
sam_scores(x)
```

Arguments

x	A geosam or geosam_image object.

Value

Numeric vector of scores.

---

sam_select *Select Specific Detections by Index*

Description

Subsets a geosam object to include only specific detections.

Usage

```
sam_select(x, index)
```

Arguments

x	A geosam object.
index	Integer vector of indices to keep.

Value

A new geosam object with only the selected detections.

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "building")
# Keep only the top 3 scoring detections
top3 <- sam_select(result, order(sam_scores(result), decreasing = TRUE)[1:3])

## End(Not run)
```

sam_unload *Unload SAM3 Model*

Description

Unloads the SAM3 model from memory to free GPU/CPU resources.

Usage

```
sam_unload()
```

Value

Invisibly returns TRUE.

sam_view *Interactive Viewer for SAM Detections*

Description

Opens an interactive map viewer to view SAM detections.

Usage

```
sam_view(
  x,
  fill = "#FACC15",
  border = "#EAB308",
  fill_opacity = 0.5,
  source = NULL
)
```

Arguments

x	A geosam object from `sam_detect()`.
fill	Fill color for detection polygons. Default is "#FACC15" (yellow).
border	Border/outline color for polygons. Default is "#EAB308".
fill_opacity	Fill opacity for detection polygons (0-1). Default is 0.5.
source	Imagery source for basemap: "mapbox", "esri", or "maptiler". If NULL (default), uses the source from the geosam object if available, otherwise falls back to "mapbox" or "esri" if no API key is set.

Details

The viewer shows the satellite imagery with current detections overlaid.

Note: Point-based refinement is not yet fully implemented. The current version is view-only.

Value

The geosam object when the user clicks "Done".

Examples

```
## Not run:
result <- sam_detect(image = "satellite.tif", text = "building")
refined <- sam_view(result)

# Custom colors
sam_view(result, fill = "#3B82F6", border = "#1D4ED8")

## End(Not run)
```

---

sam_view_image *View Detection Results on Image*

Description

Opens an interactive viewer to display detection results overlaid on the source image.

Usage

```
sam_view_image(x)
```

Arguments

x	A geosam_image object from sam_image() or sam_explore_image().

Value

The geosam_image object when the user clicks "Done".

Examples

```
## Not run:
result <- sam_image("photo.jpg", text = "dog")
sam_view_image(result)

## End(Not run)
```

Index