Object Attributes Annotation Prompt
Annotate this 3D asset assuming it can be found in an urban scenario (i.e., a street, a road, a sidewalk, a neighborhood, a garage, an park, etc), with the following values:
{
"annotations": {
"description_long": a very detailed visual description of this [category_l3] object that is no more than 6 sentences. Don't use the term "3D asset" or similar here and don't
comment on the object's orientation. Do use proper nouns when appropriate.,
"description": a 1-2 summary of description_long, keep the description rich and visual,
"description_view_ <i>": a short description of this [category_l3] object from view i (highlight/compare features that are different from other views),</i>
"category": [category_l3], "hoight": approximate height of this [category_l2labicat in maters (m). Papert the height for the chiest's orientation as shown in the images. For a standing human male this.
"height": approximate height of this [category_l3] object in meters (m). Report the height for the object's orientation as shown in the images. For a standing human male this could be "1.75",
"max_dimension": approximate maximum dimension of this [category_l3] object in meters (m). This is the longest dimension of this [category_l3] object, regardless of
orientation. This should always be greater or equal to the height,
"materials": a Python list of the materials that this [category_l3] object appears to be made of, taking into account the visible exterior and also likely interior (roughly in order
of most used material to least used; include "air" if the object interior doesn't seem completely solid),
"materials_composition": a Python list with the apparent volume mixture of the materials above (make the list sum to 1),
"mass": approximate mass of this [category_l3] object in kilogram (kg) considering typical densities for the materials. For a human being this could be "72",
"receptacle": a boolean indicating whether or not this [category_l3] object is a receptacle (e.g. a bowl, a cup, a vase, a box, a bag, etc). Return true or false with no
explanations,
"frontView": integer index of the view that represents the front of this [category_l3] object. This is typically the view from which you would approach the object to interact
with it, "quality": a number 0.10 indicating the quality of this [ category 12 ] abject 0 is very low quality (amotourish confusing missing toytures a 2D scan with many holes ato).
"quality": a number, 0-10, indicating the quality of this [category_l3] object. 0 is very low quality (amateurish, confusing, missing textures, a 3D scan with many holes, etc), 10 is very high quality (professional, detailed, etc),
"movable": a boolean indicating whether this [category_l3] object is movable or fixed/static in the environment. Return true or false with no explanations,
"required_force": an approximate force in Newtons (N) required to move or push this [category_l3] object if it is movable. Base your estimate on the object's size, shape,
and apparent material. Return 0 if the object is clearly immovable (e.g., a building or embedded structure),
"walkable": a boolean indicating whether agents (e.g., humans, robots) can walk or move on this [category_l3] object (e.g., flat roofs, wide benches). Return true or false
with no explanations,
"enterable": a boolean indicating whether agents can physically enter this [category_l3] object (e.g., vehicles, doorways, booths, houses or buildings). Return true or false
with no explanations,
"affordances": a Python list of high-level functional affordances that this [category_l3] object provides (e.g., "sittable", "openable", "closable", "pressable", "toggleable",
"drivable", "rotatable", "pushable", "pullable", "liftable"). Return only the most relevant affordances. Exclude general affordances such as "enterable", "walkable", and "movable",
as they are annotated elsewhere. Return only the list, without explanation or additional text.,
"support_surface": a boolean indicating whether this [category_l3] object can physically support other objects placed on it (e.g., tables, platforms, roofs). Return true or
false with no explanations, "interactive parts": a Python list of distinct functional parts or components of this [ cotogony 12 ] object that can be interacted with (o.g., "bandle", "drawer", "wheel", "door", "
"interactive_parts": a Python list of distinct functional parts or components of this [category_l3] object that can be interacted with (e.g., "handle", "drawer", "wheel", "door", "button"). Only include parts visible or implied from geometry. Return only the list, no explanation or other words,
" <b>traversability</b> ": a string label describing how an agent might traverse this [category_l3] object. Choose one of: "pass_through", "push_through", or "obstacle",
"traversable_by": a Python list of agent types that can traverse or pass through this [category_l3] object (e.g., "person", "wheeled_robot", "drone"). Only include agents for
which traversal is physically feasible. If the object is not traversable by any agent, return an empty list (i.e., []).,
"colors": a Python list of the visible colors of this [category_l3] object (e.g., ["white", "gray", "blue"]). Focus on dominant and distinct colors visible from the exterior,
"colors_composition": a Python list of floats representing the approximate volume composition of the colors listed above. Ensure the list sums to 1.0 and corresponds to the
order of "colors",
"surface_hardness": a string describing the tactile hardness of the surface of this [category_l3] object. Choose one of: "soft", "semi-soft", or "hard",
"surface_roughness": a float in the range [0, 1] indicating the micro-texture roughness of the surface of this [category_l3] object. 0 means perfectly smooth (e.g., polished
glass), 1 means extremely rough (e.g., coarse stone),
"surface_finish": a string describing this [category_l3] object's surface tactile/visual quality. Choose one of: "rough", "matte", "smooth", "glossy", "sleek", or "grippy",
"reflectivity": a float in the range [0, 1] that controls how much light is reflected by the surface of this [category_l3] object. 0 means no visible reflection, 1 means mirror-like
reflection,
"index_of_refraction": a float representing the surface's optical index of refraction (IOR) of this [category_l3] object. Typical values range from 1.0 (air) to ~2.5 (diamond).
Use realistic values based on material type. Higher values increase reflection and refraction at oblique angles,
"youngs_modulus": approximate material stiffness of this [category_l3] object in Megapascals (MPa). Use realistic values inferred from material types (e.g., 1e7 for rubber, 2e11 for steel),
"friction_coefficient": a positive float representing the estimated friction coefficient, based on this this [category_l3] object's material and surface finish (e.g., polished ice
$\approx$ 0.01, plastic $\approx$ 0.3, wood $\approx$ 0.5, rubber $\approx$ 0.9, dry concrete $\approx$ 1.2),
"bounciness": a float in the range [0, 1] representing the expected elasticity of the [category_l3] object upon impact. Higher values indicate more bounce (e.g., rubber ball ≈
0.9), while lower values indicate minimal or no bounce (e.g., stone ≈ 0.0),

"asset\_composition\_type": a string describing the structural nature of this 3D asset. Choose one of: "single" (a standalone atomic object), "group" (a small collection of related objects), or "scene" (a full composite scene with layout),

or interference with agents,

"recommended\_clearance": approximate safe buffer distance (in meters) that should surround this [\_\_category\_l3\_\_] object when placed in a scene. This helps avoid collision