VLM-guided Object-level Segmentation from Dynamic Scene

Abstract: Promptable segmentation has lowered the barrier to extracting pixel-accurate regions, yet current models are essentially part-level engines: they respond well to local cues but remain agnostic to *object*, frequently fragmenting a single instance into multiple masks. We present a training-free pipeline that lifts part-level predictions to object-level masks by coupling open-vocabulary semantics from a vision–language model (VLM) with a SAM2 grounding-and-masking backend. The VLM inventories the scene and returns a normalized list of object names and aliases. The labels, without any boxes, points, or hand-crafted prompts, are passed to the grounding–segmentation stack, which produces instance-consistent masks for each named object. A lightweight orchestration layer handles name canonicalization, synonym expansion, and conflict resolution (e.g., "table" versus "table leg") and consolidates fragments while preserving the boundary quality of the underlying segmenter. On a variety of everyday scenes, we show that our model can handle both real-world images as well as simulation renderings. It is sufficient to obtain object-aligned masks that are directly usable for object-centric editing and downstream reasoning, as well as practical robot perception tasks such as grasp planning and object-centric mapping. Beyond practicality, our findings argue for a clean separation of concerns—semantics from VLMs, spatial precision from promptable segmenters, as a robust, open-vocabulary front end for object-level scene understanding and a lightweight component in robotics pipelines.

Keywords: 2D Segmentation, Vision Language Model, Robotic Perception

1 Introduction

Promptable segmentation has made per-frame region extraction broadly accessible. Models such as SAM and its successors deliver strong cross-domain generalization and high-quality boundaries [1, 2]. However, the default behavior of these models is to produce a set of part-level masks: local texture and edge evidence is captured exceptionally well, but a single *object* (e.g., a chair) is often split into several pieces (seat, back, legs). For many applications—interactive editing workflows, video summarization, inventorying and counting, and object-centric analysis—systems require *object-level* masks that align with semantic identities rather than a collection of visually coherent parts. Classical instance/semantic segmentation pipelines impose objectness by design [3, 4, 5], but they typically rely on task-specific training, fixed vocabularies, and dataset curation, which limits plug-and-play use in open-world, dynamic scenes.

In parallel, vision–language models (VLMs) have become effective open-vocabulary recognizers [6, 7, 8, 9]. Given a natural image, a VLM can enumerate the objects present and provide canonical names and aliases. This suggests a simple decomposition of responsibilities: let *semantics* be produced by VLMs ("what is in the scene?"), and let *spatial precision* be delegated to a promptable segmenter ("where are the pixels?"). The central question is whether *text-only* labels, obtained from a VLM, suffice to drive object-level masks without any clicks, boxes, or precomputed prompts. We answer in the affirmative by coupling VLM semantics with a grounding-and-masking stack (Florence2 + SAM2) that accepts category names and produces per-object masks in a zero-shot

manner [10, 2]. Unlike open-vocabulary detection pipelines that still depend on proposals or explicit spatial prompts [11, 12, 13, 14], our interface remains label-only: names in, object masks out.

Concretely, we treat an input video as a sequence and first select a compact set of *keyframes* that capture salient motion and viewpoint changes. A VLM inventories and normalizes object labels on these keyframes, collapsing synonyms (e.g., `cup`/`mug`, `framed picture`/`painting`) and removing ill-posed entries. The resulting name lists are fed directly to a Florence2 + SAM2 stack that performs open-vocabulary grounding and mask extraction end-to-end; no spatial signals are produced by—or requested from—the VLM. A lightweight orchestration layer resolves conflicts (e.g., "table" versus "table leg") using coverage- and connectivity-aware rules and consolidates fragments into instance-consistent masks. The design is training-free and model-agnostic: any VLM that lists objects, and any grounding + masking pair that accepts category names, can be substituted without retraining.

A label-only, keyframe-driven interface is practical. It is able to remove per-frame prompt engineering, amortize computation across representative frames, and prevent dependencies on proposal mechanisms that are brittle across environments. For multi-object scenes it scales naturally; the VLM can emit dozens of categories essentially for free per keyframe, and the segmentation backend resolves localization internally. Most critically, separating out VLM-VFE concerns results in robustness and transparency such that failures—whether they be names (near-duplicates) or locations (very small, or heavily occluded)—are independently diagnosable and may easily be remediated with a few simple rules. Our qualitative results across everyday scenes indicate that such a label-only pipeline yields object-aligned masks while maintaining the boundary quality of the underlying segmenter, thereby serving as a practical front end for object-centric editing, counting, and downstream reasoning; when desired, these layers also plug into lightweight robotics use cases (e.g., grasp target pre-selection or teleoperation overlays). We view this as complementary to supervised instance/semantic segmentation [3, 4, 5] and referring segmentation [15, 16]: the goal is not to surpass trained baselines on closed-vocabulary benchmarks, but to provide an open-vocabulary, training-free mechanism that converts part-level predictions into object-level masks.

2 Related Work

2.1 Open-Vocabulary Segmentation

Research on open-vocabulary segmentation has grown rapidly alongside the development of promptable models such as SAM and its successors, which enable class-agnostic segmentation from points, boxes, or text while showing strong cross-domain generalization [1, 2]. On the supervised side, unified decoders such as Mask2Former remain strong baselines for semantic, instance, and panoptic segmentation [5], while classical detector-based approaches (e.g., Mask R-CNN) and transformer query methods (e.g., DETR) continue to be both competitive and interpretable [3, 4]. Beyond these closed-vocabulary systems, several works have sought to extend segmentation into the open-vocabulary regime by injecting language into pixel prediction. Examples include OpenSeg/OVSeg [17, 18], CLIP-guided approaches such as LSeg and CLIPSeg [19, 20], and grouping-based methods like GroupViT [21]. Most of these approaches impose object or category structure during training or through adaptation. In contrast, our work does not modify the segmenter at training time: semantics are introduced only at inference through a text-only interface, producing object-level masks without any additional spatial prompts.

2.2 Vision–Language Recognition and Text-to-Region Grounding

Large VLMs such as CLIP, BLIP-2, LLaVA, and Qwen-VL support open-vocabulary recognition and scene-level naming [6, 7, 8, 9]. Text-to-region grounding maps a category name to a localized region. Representative systems include Detic [11], GLIP [12], OWL-ViT [13], ViLD [22], and Grounding DINO [14]. Florence2 provides an open-world backbone and a flexible multimodal text interface [10]. Most existing methods still rely on proposals or spatial prompts. We take a text-only route instead: the VLM outputs a normalized list of labels, and the grounding–masking backend

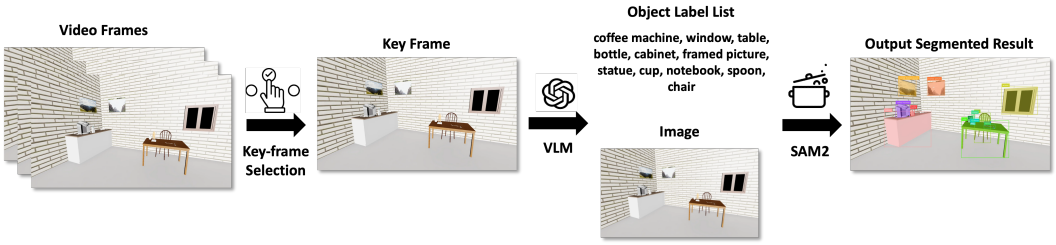

Figure 1: **Pipeline overview (text-only to masks).** (1) A VLM inventories objects from a keyframe image and produces a normalized label set through canonicalization and alias clustering. (2) Florence2 + SAM2 takes these labels as input and performs open-vocabulary grounding and mask extraction, generating candidate masks for each object. (3) A lightweight consolidation stage enforces objectness via mask-space NMS across aliases, per-label part merging, parent–part conflict resolution, and cross-label disambiguation. The system outputs per-object masks and layered visualizations, without requiring any clicks, boxes, or image-specific prompt engineering.

turns those names into per-object masks. This split between naming and masking keeps the interface simple and robust, and it reduces manual work in scenes with many objects.

2.3 Objectness, Referring Segmentation, and Part–Whole Consolidation

The idea of enforcing objectness has been central to both instance and panoptic segmentation for many years [3, 4, 5]. Referring segmentation, which maps natural language phrases to pixel regions, is particularly effective when the reference is specific [15, 16]. In open-world settings, however, the requirement is often to *discover and mask all objects* without predefined categories. Our formulation is designed with this use case in mind: objects are named by a VLM (without coordinates), and a grounding-and-masking backend produces corresponding masks. Only lightweight consolidation is applied—handling synonyms, resolving parent–part overlaps (e.g., "table" vs. "table leg"), and merging fragments—while preserving the high boundary quality of the underlying segmenter. This perspective is complementary to language-driven segmentation and open-vocabulary classification [17, 19, 20, 18, 21]: rather than training new pixel classifiers, we repurpose existing promptable segmenters and introduce semantics at inference time through a text-only interface.

3 Method

Our goal is to convert *text-only* object inventories into instance-consistent, object-level masks, without requiring clicks, boxes, or additional task-specific training. As depicted in Fig. 1, the proposed pipeline is structured into three stages: (i) a *label inventory* step, where a vision–language model (VLM) is queried to return object names and aliases, followed by normalization; (ii) a *text-to-mask* step, where Florence2 + SAM2 takes the normalized names as its only inputs and produces candidate masks for each object; and (iii) an *objectness consolidation* step, which merges fragmented parts, resolves parent–part conflicts, and disambiguates overlapping labels while maintaining boundary fidelity. This design explicitly separates *semantics* (from the VLM) and *spatial precision* (from the segmenter), remains training-free, and exposes a label-only interface.

3.1 Stage 1: Keyframe-Selected, Text-only Object Inventory

Keyframe selection. Given a video $\mathcal{V} = \{I_t\}_{t=1}^{T}$, we first select a compact index set of keyframes $\mathcal{K} = \{t_1, \ldots, t_M\} \subset \{1, \ldots, T\}$ with $M \ll T$ that capture salient appearance and viewpoint changes. In practice, we compute per-frame descriptors $\phi(t)$ (e.g., lightweight CNN features and color histograms) and perform farthest-point sampling with a minimum temporal spacing; uniform subsampling is also acceptable when ϕ is unavailable. All subsequent inventory operations are performed *per keyframe* I_{t_i}.

Name extraction. Given a keyframe image I_{t_i}, a VLM returns a set of free-text names $\tilde{\mathcal{L}}_{t_i} = \{\tilde{\ell}_1, \ldots, \tilde{\ell}_{\tilde{N}}\}$ describing objects present in the scene. To stabilize across paraphrases and near-duplicates, we apply a minimal normalization function

$$g(\tilde{\ell}) = \mathrm{canonical}\big(\mathrm{lower}(\mathrm{lemmatize}(\tilde{\ell}))\big),$$

which lowercases, lemmatizes, and maps common variants to canonical forms (e.g., `mugs`$\to$`mug`, `framed picture`$\to$`framed_picture`). The output is a normalized label inventory per keyframe, $\mathcal{L}_{t_i} = \{\ell_1, \ldots, \ell_N\}$, and a synonym multimap $\mathcal{A}_{t_i}(\ell)$ collecting aliases of ℓ.

Alias clustering and pruning. For each keyframe, we construct an undirected alias graph $\mathcal{G}_{t_i}^{\mathrm{alias}}$ whose nodes are names and edges link tokens with high lexical similarity or explicit coreference from the VLM. Each connected component collapses to a single canonical ℓ chosen by frequency or a language-score prior. Extremely generic entries (`object`, `stuff`) and negations are discarded. The per-keyframe inventories $\{\mathcal{L}_{t_i}\}_{t_i \in \mathcal{K}}$ are the *only* signals passed downstream; no spatial hints, points, or boxes are produced at this stage. Stage 2 then consumes the pairs $(I_{t_i}, \mathcal{L}_{t_i})$. For batch processing across the video, we optionally maintain a global union $\mathcal{L}^\star = \bigcup_{t_i \in \mathcal{K}} \mathcal{L}_{t_i}$ and an aggregated alias multimap $\mathcal{A}^\star$ for consistent naming across keyframes.

3.2 Stage 2: Text-to-Mask Orchestration (Florence2 + SAM2)

Grounding-and-masking contract. We denote by $\mathcal{G}\&\mathcal{M}$ a grounding-and-masking stack that accepts a name list and an image and returns candidate instance masks per label:

$$\mathcal{C} \;=\; \mathcal{G}\&\mathcal{M}(I, \mathcal{L}) \;=\; \big\{(\ell, M_\ell^{(k)}, s_\ell^{(k)})\big\}_{\ell \in \mathcal{L},\, k=1\ldots K_\ell},$$

where $M_\ell^{(k)} \in \{0,1\}^{H \times W}$ and $s_\ell^{(k)} \in [0,1]$ is a confidence. In our instantiation, Florence2 provides open-vocabulary grounding conditioned on the text ℓ and internally yields spatial priors that SAM2 converts into pixel masks with high boundary fidelity. Crucially, the interface remains *text-only*: names are the sole inputs exposed by our system; any spatial priors are produced within $\mathcal{G}\&\mathcal{M}$.

Multi-name fusion. Aliases $\mathcal{A}(\ell)$ may produce overlapping candidates for the same real-world object. We aggregate candidates across ℓ and $\mathcal{A}(\ell)$ by non-maximum suppression in mask space using the generalized IoU:

$$\mathrm{gIoU}(M_1, M_2) = \frac{|M_1 \cap M_2|}{|M_1 \cup M_2|} - \frac{|\, C(M_1, M_2) \setminus (M_1 \cup M_2)\,|}{|C(M_1, M_2)|},$$

where $C(M_1, M_2)$ is the smallest enclosing rectangle of the two masks. Masks with $\mathrm{gIoU} > \tau_{\mathrm{nms}}$ are merged (pixelwise max), and the new score is the geometric mean of confidences; a small morphological closing stitches micro-gaps.

3.3 Stage 3: Objectness Consolidation

The candidates $\mathcal{C}$ are high-quality but may still be fragmented (multiple parts of one object), conflicting (parent vs. part labels), or duplicated. We enforce objectness with three operations.

(a) Part merging within a label. For each label ℓ, we build a graph $\mathcal{G}_\ell$ whose nodes are masks $\{M_\ell^{(k)}\}$ and edges connect pairs with sufficient spatial affinity. The affinity between M_i and M_j combines overlap, boundary contact and proximity:

$$w_{ij} \;=\; \alpha\, \mathrm{IoU}(M_i, M_j) \;+\; \beta\, \frac{\mathrm{Perim}(M_i \cap \mathcal{D}_\delta(M_j))}{\mathrm{Perim}(M_i)} \;+\; \gamma\, e^{-\frac{d(M_i, M_j)}{\sigma}},$$

where $\mathcal{D}_\delta(\cdot)$ dilates by δ pixels and $d(\cdot,\cdot)$ is centroid distance. Connected components under $w_{ij} \geq \tau_{\mathrm{aff}}$ are merged (pixelwise OR). If the union of merged masks contains multiple connected components, each component is returned as a distinct instance.

(b) Parent–part conflict resolution. When two labels are in an implicit parent–part relation (e.g., `table` vs. `table_leg`) and overlap, we allocate pixels to the more specific label by a coverage test:

$$\mathrm{cov}(M_{\mathrm{part}}, M_{\mathrm{parent}}) = \frac{|M_{\mathrm{part}} \cap M_{\mathrm{parent}}|}{|M_{\mathrm{part}}|}.$$

If $\mathrm{cov} > \gamma_{\mathrm{part}}$ we retain M_{part} and remove its pixels from the parent; otherwise we keep the parent and drop the spurious part. Specificity can be approximated by token heuristics (compound tokens treated as more specific) or a label-priority list.

(c) Cross-label disambiguation. Residual overlaps across unrelated labels are resolved by per-pixel scores. Let $S_\ell(x)$ denote a soft score; if $\mathcal{G}\&\mathcal{M}$ does not expose per-pixel logits we set $S_\ell(x) = \lambda\, \hat{s}_\ell^{(j)} \mathbf{1}_{M_\ell^{(j)}}(x) + (1-\lambda)\, \kappa(x; M_\ell^{(j)})$, where κ is a normalized distance-to-center prior and $\lambda \in [0,1]$. Pixels are assigned to $\arg\max_\ell S_\ell(x)$, with ties broken by a specificity prior; islands smaller than $\eta_{\min}$ pixels are removed.

Confidence and ranking. Each instance $M_\ell^{(j)}$ is assigned a consolidated score

$$\hat{s}_\ell^{(j)} = \Big(\prod_{k \in \mathcal{K}(j)} s_\ell^{(k)} \Big)^{1/|\mathcal{K}(j)|} \cdot \rho\big(M_\ell^{(j)}\big),$$

where $\mathcal{K}(j)$ indexes candidates merged into the j^{th} instance and $\rho(\cdot)$ penalizes excessive thinness or holes (shape regularity). Low-confidence instances may be filtered or flagged as uncertain overlays.

3.4 Complexity and Implementation Notes

Notation recap: $\Theta = \{\tau_{\text{nms}}, \tau_{\text{aff}}, \gamma_{\text{part}}, \eta_{\min}, \delta, \alpha, \beta, \gamma\}$; $\mathrm{Perim}(\cdot)$ denotes 8-connected perimeter; $\mathcal{D}_\delta(\cdot)$ is binary dilation by δ pixels; $d(\cdot,\cdot)$ is Euclidean centroid distance.

The pipeline is linear in the number of labels and masks. Alias clustering and NMS are $\mathcal{O}(|\mathcal{L}| + \sum_\ell K_\ell^2)$ in practice with small K_ℓ. Graph-based merging is near-linear in edges given local connectivity. Our reference implementation uses Python with `numpy` and `scipy`; morphological operations use `skimage`. Defaults: τ_{nms}=0.6, δ=3px, (α,β,γ)=$(0.5, 0.3, 0.2)$, τ_{aff}=0.4, γ_{part}=0.6, $\eta_{\min}$=64px. We export per-object PNG masks (or COCO RLE) and a layered RGBA visualization.

3.5 Robustness and Failure Taxonomy

Near-duplicate labels (e.g., `cup`/`mug`) are handled with alias clustering. If merging is too aggressive, fine distinctions can be lost. *Small or heavily occluded objects* are often missed by the grounder; in these cases we favor precision over recall. *Objects that touch or adhere* (for example, stacked books) can trigger cross-label conflicts. Specificity rules and simple shape priors help, but they do not solve every case. Light human edits can clean up the remaining errors without changing the overall pipeline.

4 Experiments

In this study, we evaluate the proposed text-only pipeline from two complementary perspectives: (A) the *VLM inventory view*, which records the object names produced by the VLM for each image, and (B) the *text-to-mask view*, in which the pipeline feeds these names into the Florence2 + SAM2 branch to generate object-level masks. Across both views, our emphasis is qualitative: we examine whether consistent object masks are produced under different conditions. In addition, we provide a lightweight quantitative summary in terms of binary success rates.

4.1 Setup and Data

Images. We test on two sources: (i) *real-world photographs* covering indoor tabletop, household scenes, and simple outdoor snapshots; and (ii) *simulation renders* with clean geometry and con-

Algorithm 1 Text-only object-level segmentation. Symbolic pseudocode with no descriptive lines inside the body. Names from a VLM are normalized and passed to a grounding-and-masking backend (Florence2 + SAM2). Candidate masks are fused across aliases, merged by spatial affinity, and disambiguated across labels (parent–part and cross-label). Instances are scored and exported. Defaults and thresholds are given in Sec. 3.

I, VLM, $\mathcal{G}\&\mathcal{M}$, Θ
$\tilde{\mathcal{L}} \leftarrow \mathrm{VLM}(I)$; $\mathcal{L}, \mathcal{A} \leftarrow \mathrm{NormalizeAlias}(\tilde{\mathcal{L}})$
$\mathcal{C} \leftarrow \mathcal{G}\&\mathcal{M}(I, \mathcal{L})$ ▷ $(\ell, M_\ell^{(k)}, s_\ell^{(k)})$
$\mathcal{C} \leftarrow \text{Mask-NMS-Merge}(\mathcal{C}, \mathcal{A};\, \tau_{\mathrm{nms}})$
for $\ell \in \mathcal{L}$ **do**
    $\mathcal{G}_\ell \leftarrow \mathrm{BuildGraph}(\{M_\ell^{(k)}\};\, \alpha, \beta, \gamma)$
    $\{U_\ell^{(m)}\} \leftarrow \mathrm{ComponentMerge}(\mathcal{G}_\ell;\, \tau_{\mathrm{aff}})$
    $\{M_\ell^{(j)}\} \leftarrow \mathrm{SplitByConnectivity}(\{U_\ell^{(m)}\})$
$\{M_\ell^{(j)}\} \leftarrow \mathrm{ParentPartResolve}(\{M_\ell^{(j)}\};\, \gamma_{\mathrm{part}})$
$\{M_\ell^{(j)}\} \leftarrow \mathrm{CrossLabelArgmax}(\{M_\ell^{(j)}\};\, S_\ell)$
$\{M_\ell^{(j)}, \hat{s}_\ell^{(j)}\} \leftarrow \mathrm{ScoreAndFilter}(\{M_\ell^{(j)}\};\, \eta_{\min})$
$\mathrm{Export}(\{M_\ell^{(j)}\})$

trolled lighting. Representative examples include cluttered desks (cups, mugs, laptops, notebooks, spoons), kitchen corners (cabinets, coffee machines), and living rooms (windows, framed pictures, sofas), as well as simulated rooms with canonical furniture and small props.

Pipeline instantiation. Stage 1 uses a VLM to inventory objects; names are normalized and alias-clustered as in Sec. 3. Stage 2 passes names directly to a Florence2 + SAM2 grounding-and-masking stack that produces per-object masks without boxes/points. Consolidation applies mask-space NMS across aliases, per-label part merging, and parent–part disambiguation (e.g., `table` vs. `table_leg`) using the defaults given in Sec. 3. We export per-object PNG masks and layered RGBA composites for visualization.

Views reported. For each image we show: (A) the VLM name list (the semantics-only view) and (B) the corresponding masks (the spatialized view). The former reflects naming coverage; the latter tests whether names alone suffice to drive accurate localization.

4.2 Qualitative Results

Figures 2 and 3 present a qualitative gallery across real and simulated scenes. In real photographs (Fig. 2), the pipeline reliably converts VLM name inventories into object-aligned masks: `cup`/`mug`, `notebook`, `spoon`, `coffee_machine`, `cabinet`, `window`, and `framed_picture` are recovered with clean boundaries and minimal overreach. Single objects that would otherwise appear as multiple part-level regions (e.g., chair back and seat) are consolidated into a single instance per name. In simulation (Fig. 3), where geometry and lighting are more regular, the text-only interface performs similarly or slightly better; masks exhibit tight alignment to object contours and stable separation between adjacent instances.

Ablative observations. (1) *Alias handling* proves essential in practice. Treating `cup` and `mug` or `framed picture` and `painting` as equivalent reduces duplicate masks and avoids visual clutter, leading to cleaner results. (2) *Parent–part disambiguation* (e.g., distinguishing `table` from `table_leg`) prevents larger parent objects from absorbing their parts, producing more meaningful layers. (3) Applying *mask-space NMS* across aliases removes redundant small instances, and a simple morphological closing step helps to repair the small gaps left by alias redundancy.

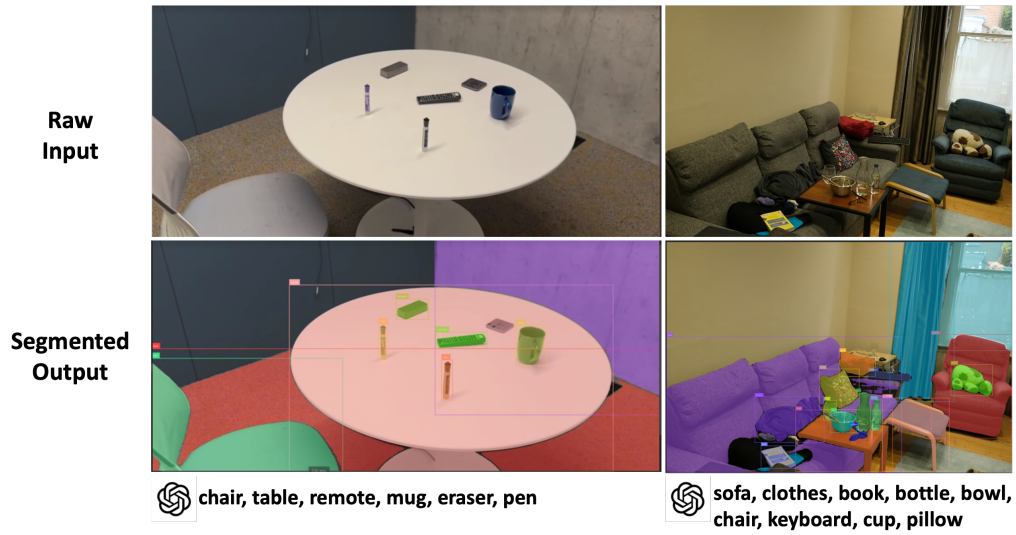

Figure 2: **Real scenes.** VLM inventory (names only) and the corresponding object-level masks from Florence2 + SAM2. The pipeline enforces objectness while preserving boundary quality. Panels are arranged as original image (top) and masked result (bottom).

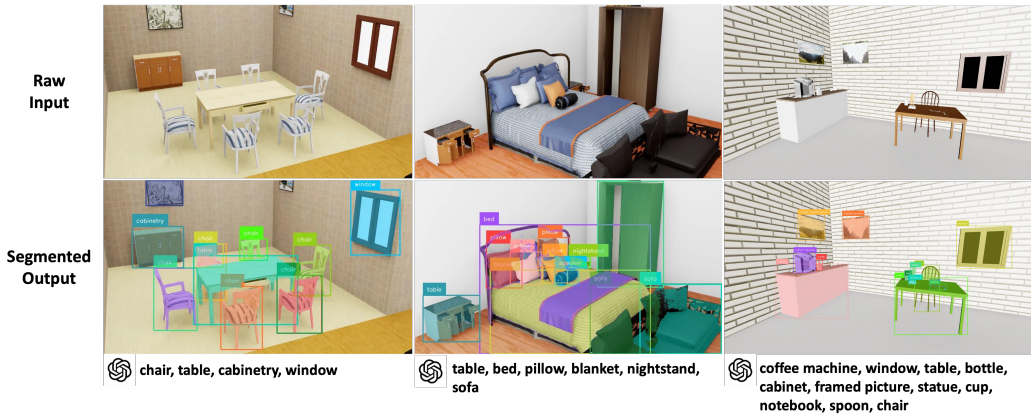

Figure 3: **Simulation scenes.** Text-only names produced by the VLM are sufficient to drive accurate masks in synthetic environments with clean geometry and lighting. Panels are arranged as original image (top) and masked result (bottom).

4.3 Quantitative Summary

Although our study is qualitative by design, we report a binary success rate to summarize overall reliability on the two splits. A case is counted as success when the target object name appears in the VLM inventory and the resulting mask is judged visually correct by human inspection (i.e., the mask aligns with the object, with no major over/under-coverage). Table 1 reports the aggregate rates.

4.4 Failure Analysis

The dominant failure mode is *missed naming*: if a category is not recognized by the VLM (often due to weak texture, dark/low-contrast surfaces, or atypical viewpoints), the downstream stack receives no label and thus produces no mask. When names are present but masks fail, causes include (i) extremely small or thin objects (e.g., pens, cables) where grounding is brittle, (ii) heavy adhesion between adjacent instances (e.g., stacked books) that yields cross-label ties, and (iii) near-duplicate categories (e.g., `cup` vs. `mug`) where aggressive alias collapsing may merge distinct fine-grained

Split	Ours (%)	SAM2 (%)
Real-world photographs	**81.4**	25.0
Simulation renders	**86.8**	20.0

Table 1: **Binary success rates.** A case is successful if the VLM inventory lists the object and the produced mask is visually correct. Success Rate is calculated by how many objects are recognized and segmented correctly across the evaluation split. The higher rate in simulation reflects more regular geometry and lighting; real photographs are affected by clutter, texture ambiguity, and illumination variance.

classes. In real photographs these issues are exacerbated by clutter and mixed lighting; in simulation they are rarer but still occur for very small props.

4.5 Diagnostics and Practical Guidance

In practice, we find that the overall recall is driven mainly by the coverage of the VLM's output list. Allowing the inventory to include a few more familiar aliases (for example treating `cup` and `mug` as interchangeable) often improves the quality of downstream masks without introducing extra effort from the user. When parent and part labels overlap, giving preference to the more specific label helps to reduce unwanted bleeding across regions. It is also helpful to apply simple shape-based rules: very small fragments below $\eta_{\min}$ can be filtered out, which keeps the final layers cleaner while discarding little useful information. With these modest adjustments, the default parameters described in Sec. 3 have been sufficient to achieve near real-time performance on ordinary hardware, making the system a practical tool for object-centric editing, counting, and lightweight perception tasks.

5 Conclusion

This work targets the gap between part-level outputs from promptable segmentation and the object-level layers needed in practice. We separate *semantics* from *spatial precision*: a VLM produces a normalized label list, Florence2 + SAM2 converts names to candidate masks, and a light consolidation step resolves aliases and parent–part conflicts. For video, we first pick representative *keyframes* and then apply this text-only, training-free, model-agnostic interface. No clicks, boxes, or prompt engineering are required.

Qualitative results on real images and simulation show that the label-only contract produces masks that align with human perception while preserving boundary quality (Fig. 2, Fig. 3). A small quantitative summary suggests stable behavior in practice (81.4% on real scenes and 86.8% in simulation; Tab. 1). The takeaway is simple: let the VLM name the objects, let the segmenter handle pixels, and keep a thin layer for objectness.

The system is already useful as a front end for editing, inventory, counting, and object-centric analysis. It also fits lightweight robotics uses such as keyframe-based perception for manipulation, pre-selecting grasp targets, building object-centric maps, and teleoperation overlays. The implementation is compact and uses standard components (mask-space NMS, basic morphology, and graph connectivity), so it is easy to reproduce and extend.

For future work, we see three directions. First, raise recall by improving the inventory (structured prompts, synonym expansion, and uncertainty-aware filtering). Second, add temporal linking so the method can generate object layers across frames and make use of SAM2 memory. Third, support quick interactive fixes (single-click relabels or merges) and feed those edits back into consolidation. These steps would make the pipeline more flexible and a better fit for robotics pipelines, including in-the-loop perception, operator-assisted teleoperation, and cost-effective sim-to-real curation.

6 Limitations

Despite its practicality, our approach has several limitations. First, it depends heavily on VLM recall: if a category is not named, no mask will be produced. This is most common for small, thin, or low-contrast objects. Second, semantic granularity can be tricky—aliases like `cup/mug` or parent–part pairs such as `table` vs. `table_leg` sometimes collapse or fragment labels. Third, difficult cases such as occlusion, adhesion (e.g., stacked books), or reflective/transparent materials remain challenging, and our simple heuristics (mask-space NMS, small-island removal) are not always robust across domains. Finally, inference time grows with the number of labels, since each triggers a grounding–masking pass.

These issues are consistent with the lightweight, text-only nature of the system. Many can be mitigated in practice by allowing longer inventories, using alias maps, or applying occasional human corrections. We view the pipeline less as a final solution and more as a practical front end—fast, training-free, and easy to extend with temporal linking, 3D lifting, or other refinements when needed.

Acknowledgments

If a paper is accepted, the final camera-ready version should include acknowledgments (e.g., to colleagues, reviewers, funding agencies, and corporate sponsors).

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, and et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[2] N. Ravi, A. Kirillov, and et al. Segment anything in images and videos (sam 2). *arXiv preprint arXiv:2408.00714*, 2024.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[4] N. Carion, F. Massa, G. Synnaeve, and et al. End-to-end object detection with transformers. In *ECCV*, 2020.

[5] B. Cheng, I. Misra, A. Schwing, R. Girdhar, and A. Kirillov. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[6] A. Radford, J. W. Kim, C. Hallacy, and et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[7] J. Li, D. Li, C. Xiong, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[8] H. Liu, C. Li, Q. Wu, and Y. J. Lee. LLaVA: Large language-and-vision assistant. *arXiv preprint arXiv:2304.08485*, 2023.

[9] J. Bai, S. Bai, and et al. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[10] L. Yuan, C. Cui, H. Zhang, et al. Florence-2: Advancing a vision foundation model for open-world understanding. *arXiv preprint arXiv:2306.03046*, 2023.

[11] X. Zhou, R. Girdhar, A. Joulin, S. Mello, and P. Krähenbühl. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.

[12] L. H. Li, P. Li, M. Yatskar, and et al. GLIP: Grounded language-image pre-training for open-vocabulary object detection. In *CVPR*, 2022.

[13] M. Minderer, X. Zhai, L. Beyer, and et al. OWL-ViT: Open-vocabulary object detection via vision transformers. In *CVPR*, 2023.

[14] S. Liu, Z. Zeng, F. Li, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv:2303.05499*, 2023.

[15] L. Yu, P. Poirson, S. Yang, and et al. Modeling context in referring expressions. In *ECCV*, 2016.

[16] S. Yang, Z. Li, S. Li, and et al. LAVT: Language-aware vision transformer for referring image segmentation. In *CVPR*, 2022.

[17] G. Ghiasi, X. Gu, Y. Cui, and et al. Open-vocabulary image segmentation. In *CVPR*, 2021.

[18] Y. Liang, Z. Wang, T. Wu, and et al. OVSeg: Open-vocabulary semantic segmentation with masks and text. *arXiv preprint arXiv:2304.04975*, 2023.

[19] B. Li, W. Yin, and et al. Language-driven semantic segmentation. In *CVPR*, 2022.

[20] T. Lüddecke and A. Ecker. CLIPSeg: Image segmentation using text and image prompts. In *CVPR*, 2022.

[21] Y. Xu, S. Wang, K. Zhang, and et al. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.

[22] X. Gu, T.-Y. Lin, W. Kuo, and et al. ViLD: Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.