

Hierarchical Semantic Segmentation with Autoregressive Language Modeling

Josh Myers-Dean¹, Brian Price², Yifei Fan², Danna Gurari^{1,3}

¹University of Colorado Boulder, ²Adobe Research, ³The University of Texas at Austin

Abstract

Hierarchical semantic segmentation entails progressively decomposing objects into smaller nested parts. Existing approaches either require multiple inference passes or multiple, fixed decoders. We instead introduce HALLUMI, an autoregressive language modeling framework that performs the task in one inference pass, relying on special tokens to indicate parent-child relationships so the hierarchy can be recovered from the generated text. Experiments on a hierarchical semantic segmentation dataset to the subpart-level (SPIN) show HALLUMI achieves state-of-the-art results.

1. Introduction

While we are witnessing exciting progress in computer vision, a lingering limitation of modern models is their lack of a core aspect of visual intelligence: *understanding how objects are progressively decomposed into recursively more primitive, nested parts until further decomposition is not possible* [2, 13, 50, 56]. In this work, we focus on hierarchical decomposition to the *subpart* granularity; specifically, given an image, the task entails segmenting and naming all objects, their parts, and those parts’ nested parts (i.e., subparts). Called *hierarchical semantic segmentation*, this task is fundamental to analyzing an object’s inherent hierarchical structure, whether it is human-made (e.g., a car has a wheel composed of a tire, rim, and hub) or natural (e.g., a bird has a head with eyes).

Currently, only three models are designed for hierarchical semantic segmentation to the subpart granularity or beyond [28, 49, 57]. Two of these models, HIPIE [57] and ViRReq [49], require *multiple inference passes* to recover a whole hierarchy; HIPIE uses distinct prompts and decoders for predicting objects and their (sub)parts separately, while ViRReq requires independent inference passes at each level of granularity. This is exemplified in **Figure 1(b)**. The third model, HSSN [28] relies on a *multilabel classification framework* to associate pixels with multiple labels at different hierarchy levels. This last design is not only difficult to scale (i.e., the output space can grow exponentially with

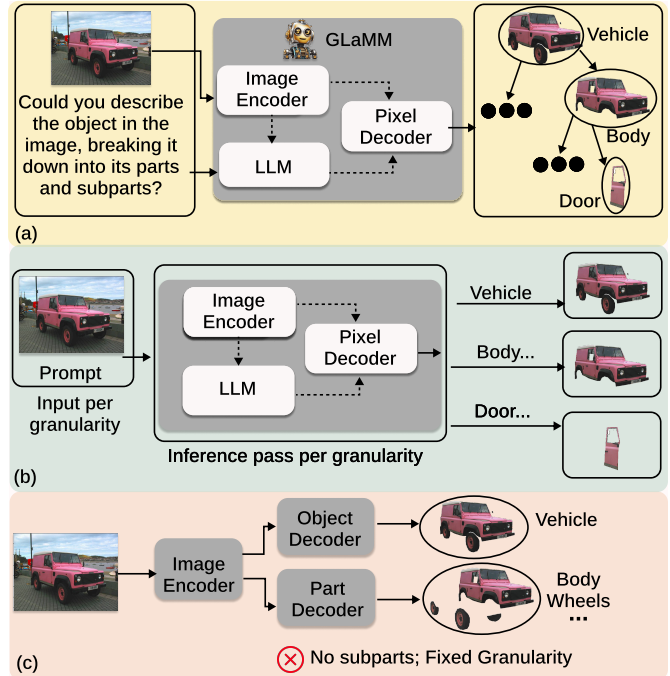


Figure 1. High-level overview of hierarchical semantic segmentation approaches: (a) our proposed HALLUMI framework, (b) multiple decoder head framework, and (c) multiple inference pass framework. Only HALLUMI can segment objects, parts, and subparts in a single inference pass while explicitly specifying hierarchical relationships. We omit most parts and subparts for clarity.

increased hierarchy depth¹), but also necessitates extra post-processing to (1) enforce that exactly one class can be predicted per hierarchy level and (2) prevent finer-grained entities from being predicted in the absence of coarser-grained parent entities. Our work is motivated by the belief that all existing approaches suffer from unnecessary complexity. We aim to streamline the pipeline such that only a single inference pass is needed, without further post-processing.

One way to streamline the process could be to use multiple parallel decoder heads to generate multiple segmentations at differing granularity levels [40], similar to the popu-

¹HSSNs output space must span object, part, and subpart classes, so it grows quickly as the hierarchy deepens, increasing computational costs.

lar DETR-based [3] and SAM-based [19, 45] segmentation methods. This is exemplified in **Figure 1(c)**. Yet, such a design is problematic, as it requires hard-coding the number of decoder heads. This hard-coding can be either computationally wasteful or insufficient, depending on whether that number is more or less than the number of target entities in the image [25, 57]. Additionally, because this approach necessitates a separate decoder for each granularity, it would require architectural changes to scale to deeper hierarchies, alongside larger memory requirements.

We instead propose an *autoregressive language modeling framework* that performs hierarchical semantic segmentation in a single inference pass. Our approach offers four key advantages. First, it can generate a caption of *arbitrary length* alongside segmentations indicating where each phrase appears. Second, it uses special tokens to mark parent-child relationships, so that the hierarchy can be recovered without post-processing. Third, it follows a breadth-first strategy, describing coarser entities before finer ones to guide predictions (for example, predicting “handle” only after “door”). Coarser entities are generally easier to recognize, and so can provide a valuable cue to large language models about which children likely follow. Fourth, it relies on a single segmentation decoder head, allowing it to scale linearly with sequence length and easily accommodate hierarchies deeper than subparts. Unlike existing methods that rely on a fixed number of decoders or multiple inference passes, our approach performs explicit hierarchical recovery using only one pixel decoder and one inference pass, as exemplified in **Figure 1(a)**.

We call our approach **HALLUMI**, which is short for **H**ierarchical **A**utoregressive **L**ocalized **L**angUage **M**odelIng. We implement HALLUMI by extending GLaMM [43], a grounded conversation generation model, with task-specific tokens and a task-specific training protocol. To our knowledge, HALLUMI is the *first model that performs hierarchical semantic segmentation in a single inference pass*. Experiments demonstrate HALLUMI outperforms state-of-the-art models overall, and especially at the subpart granularity. More generally, its success can benefit many applications with a need to recognize and segment entities at different granularity levels. Examples include image/video retrieval, image/video editing, automatic magnification, tactile discovery, robotics, and manufacturing.

2. Related Work

Hierarchical Segmentation Models. Since the early 2000s, hierarchical segmentation methods have been proposed. Initially, they were called “deformable part” [10, 11] and “image parsing” [12, 14, 48, 52, 66, 68] models and later adopted numerous names in the deep learning era [7, 16, 25, 29, 30, 61–63]. However, most methods are not suitable for our paper’s focus because they focus only on

predicting objects and their parts (and occasionally also a parent scene type, such as “kitchen”). The three methods that support hierarchical decomposition to the subpart granularity [28, 49, 57], as discussed in Section 1, have complex designs. We introduce the first streamlined pipeline for producing a full hierarchy in a *single inference pass* without further post-processing, by posing an autoregressive language modeling framework.

Image Captioning Models. Our work extends the latest advancements in image captioning models, particularly those utilizing multimodal transformers [9, 54]. Many can autoregressively ground entities, but either only produce segmentations with no associated caption [24, 55, 60], produce captions with only bounding box detections [38, 39, 64], or produce only bounding box detections [6]. While bounding box-based models *can* be combined with promptable segmentation models such as SAM [19] (e.g., [45]) to generate masks, such integration introduces additional complexity that prevents end-to-end training.

Only two models enable pixel-level localization directly from special tokens in an image caption: GLaMM [43] and GROUNDHOG [65]. However, these models are designed for object-centric recognition and segmentation, comprehensively describing the entities within a *scene*. We propose to similarly follow this framing of segmentation as a Grounded Conversation Generation [43] task, outputting a comprehensive description with interleaved segmentation masks. Unlike prior work, we address a more challenging task of recognizing and localizing recursively nested entities of increasingly smaller sizes for finer-grained entities.

3. Method

We now introduce our HALLUMI framework for hierarchical semantic segmentation, summarized in **Figure 2**. It consists of four key design decisions about the (1) model architecture, (2) model vocabulary, (3) training approach, and (4) training data.

Model Architecture. We employ an autoregressive framework that predicts each token x_t at time step t based on prior tokens in the sequence. For a target sequence $X_{1:L}$ of length L , the probability of generating this sequence can be formalized using the chain rule of probability:

$$p(X_{1:L} \mid I, P, \theta) \propto \prod_{t=1}^L p_{\theta}(x_t \mid I, P, x_{<t}) \quad (1)$$

where I is an encoded image, P is a prompt (e.g., instructions), and θ are the model weights. In other words, the probability of generating a target sequence is proportional to the product of the probabilities of generating each token

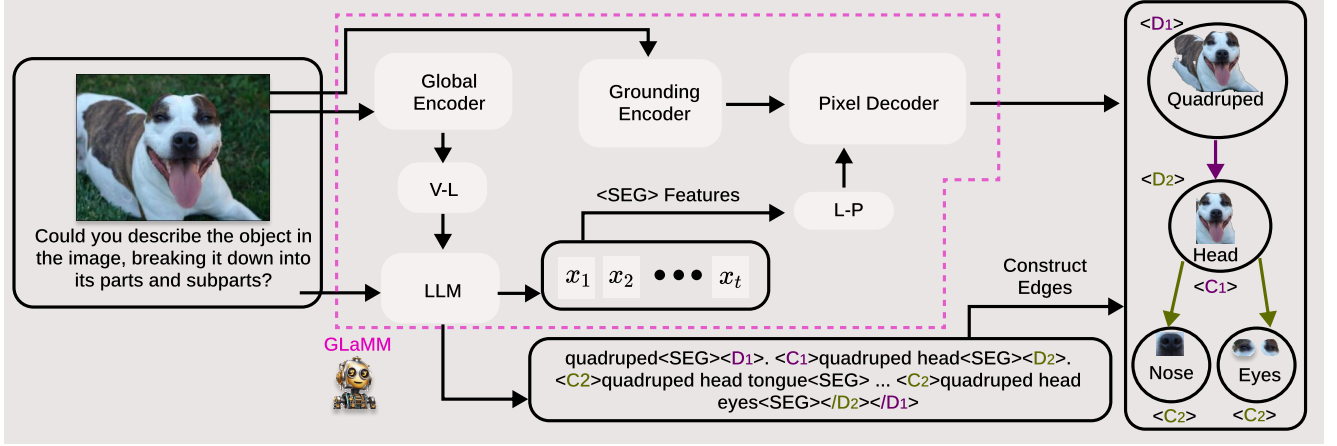


Figure 2. Overview of HALLUMI. Our method unifies autoregressive text generation and segmentation using a single mask decoder triggered by $\langle \text{SEG} \rangle$ token features. The model vocabulary includes $\langle p \rangle$, $\langle /p \rangle$, and $\langle \text{SEG} \rangle$ to mark segments, and $\langle D_i \rangle$, $\langle /D_i \rangle$, and $\langle C_i \rangle$ to encode explicit parent-child relationships. Captions are generated breadth-first to systematically describe and segment objects, parts, and finally subparts. We implement HALLUMI using GLaMM, a vision-language model with SAM-based decoding, in a single end-to-end pipeline. Most parts and subparts are omitted here for clarity.

x_t , which depends on the image I , the prompt P , and all preceding tokens $x_{<t}$.

Critically, the framework includes a decoder module to generate segmentations for all phrases paired with special tokens (e.g., $\langle \text{SEG} \rangle$), as described below. In our case, for each $\langle \text{SEG} \rangle$ token, the decoder takes as input its features and then outputs a binary mask. *Since our captions already assign semantic categories and the concept of a segmentation is granularity-agnostic, only a single mask decoder is needed.* While captions are produced autoregressively, the masks can be decoded in parallel after all text is generated.

Model Vocabulary. Our approach uses tokens in the model’s vocabulary to explicitly convey hierarchical structures, as shown in **Figure 2**.

Following earlier work [24, 43], we use $\langle p \rangle$ and $\langle /p \rangle$ tokens to define the span of text to be segmented and a $\langle \text{SEG} \rangle$ token to mark which embedding from the LLMs final hidden states should be decoded into a segmentation. For example, given the token sequence $\langle p \rangle$ A man $\langle /p \rangle \langle \text{SEG} \rangle$ sits on $\langle p \rangle$ an old, red bench $\langle /p \rangle \langle \text{SEG} \rangle$, each $\langle \text{SEG} \rangle$ token prompts the mask decoder module to segment the preceding phrase. This design follows the embedding-as-mask paradigm first introduced for semantic segmentation in 2023 [24], where localization is posed as a language modeling task driven by special tokens.

We also introduce three tokens that clarify parent-child relationships. Two of these tokens, $\langle D_i \rangle$ and $\langle /D_i \rangle$, are postfix tokens (placed after $\langle \text{SEG} \rangle$), indicating whether the i^{th} entity can be decomposed further. The third token, $\langle C_i \rangle$, is a prefix token (placed before the entity to be grounded)

that specifies parent-child relationships, telling the model that the current entity is a child of the i^{th} entity. By matching $\langle C_i \rangle$ to the corresponding $\langle D_i \rangle$ token, the hierarchy can be recovered without post-processing heuristics. This explicit structure improves interpretability by showing which segments require further decomposition and how new content relates to earlier segments.

Training Approach. We fine-tune a pre-trained model that meets our architecture requirements using instruction tuning [32, 58, 67]. This is the key ingredient that enables the model to know when to perform our task, rather than other tasks it supports, by clarifying to perform our task only when it sees a specific instruction; e.g., [SYSTEM PROMPT]; USER: "Could you describe the object in the image, breaking it down into its parts and subparts and providing interleaved segmentation masks for both the object and interleaved parts and subparts?" Input [X]; ASSISTANT:. We prepend the same, pre-defined instruction to each training example (described below) to match what will be seen at test time.

We use LoRA [18] for fine-tuning since it enables us to keep the model frozen by training extra task-specific parameters that steer how the pretrained model performs the target task. This approach offers several training benefits, including faster optimization and reduced memory requirements due to the limited number of trainable parameters. It also brings inference benefits, such as reducing memory requirements since no new components must be deployed. Standard captioning and semantic segmentation losses are

used during training to optimize the model’s ability to generate accurate natural language captions and segmentations.

Training Data Generation. Our training data creation process assumes a supervised learning approach where we have access to a labeled dataset providing hierarchical semantic segmentations to the (sub)part granularity. For each ground truth example, we use all entity segmentations as is and embed their associated semantic labels into a structured caption template. The motivation for structured templates is to mitigate potential impacts of the domain shift we suspect exists between conventional natural language captions (e.g., commonly found in Internet text) and the (awkward-sounding) captions that progressively convey all hierarchical details of objects. Constraining captions to follow a single structure can aid the model’s learning by limiting the linguistic diversity it must learn [22, 35, 37].

For our caption template, we use a *breadth-first* approach with the following structure²: “The <image> contains <p> [object 1] </p> <SEG> . <p> [object 1] [part 1] </p> <SEG> <p> [object 1] [part 2] </p> <SEG> . The <p> [object 1] [part 1] [subpart 1] </p> <SEG> ...”, with periods denoting the end of a given hierarchy level. While our captions lack natural language flow, they belong to a context-free language³, and thus are machine comprehensible [47]. When authoring nested parts and subparts, we prepend the context from the coarser entities (e.g., referring to “eyes” as “quadruped head eyes”) since we found from our experiments (described in Section 4.3) that this yields better performance.

Implementation. While we intend our approach to be model-agnostic, at the time of this paper’s submission, only one model was publicly available that met our architecture requirements: GLaMM [43].⁴ Introduced in 2024, it supports end-to-end training for grounded text generation and supports a variety of captioning and segmentation tasks.

GLaMM relies on three key components: a large language model (LLM) (Llama 2 [51]) for language processing and generation, a grounding image encoder (ViT-H [9]) to generate an image representation suitable for segmentation, and a mask decoder (SAM [19]) to generate segmentation masks from learned <SEG> token embeddings. GLaMM also incorporates “glue” layers to facilitate interactions between vision and language features: vision-language projection layers integrate image features within the LLM, and a language-to-prompt layer translates LLM outputs that cor-

respond to the <SEG> tokens into prompts for SAMs mask decoder. As input to the model, a global image encoder (CLIP [41]) generates the embedding passed which gets passed in “image” as part of the input prompt. Additionally, the model takes as input an optional region encoder for user-provided visual prompts (e.g., bounding boxes).

We fine-tune GLaMM using pre-trained weights from the variant trained on Grand [43]. Following GLaMM’s fine-tuning recipe, we fully fine-tune the vision-to-language and language-to-prompt projection layers, as well as the pixel decoder, and use LoRA [18] with a rank of 8 to fine-tune the LLM. This results in fine-tuning approximately 5% of GLaMM’s parameters. We use GLaMM’s loss functions, which minimize cross-entropy for the textual outputs as well as a linear combination of Dice loss [34] and binary cross-entropy for the segmentation outputs. We use a batch size of 32 and train for 1,700 iterations across 8 NVIDIA A100 GPUs using DeepSpeed [44] stage 2 optimization.

4. Experiments

We now evaluate our HALLUMI implementation and compare it to modern models also capable of outputting hierarchical semantic segmentations in Section 4.2, with five baseline model variants for the task of hierarchical segmentation to the subpart granularity and six baseline models for the task of hierarchical segmentation to the part granularity. We also provide fine-grained analysis on the design choices for HALLUMI to underscore their benefit (Section 4.3).

4.1. Experimental Design

Dataset. We benchmark all methods on the SPIN dataset [36]. This dataset uniquely supports analyzing performance for deeper hierarchy levels, as it is the only publicly-available option providing hierarchical semantic segmentations to the *subpart* granularity. SPIN is based on PartImageNet [15] and contains one salient region, 11 object categories, 40 part categories (34 of which have subpart decompositions), and 203 subpart categories across 10,387 images. We use the standard splits: 8,828 images for training, 519 for validation, and 1,040 for testing.

Evaluation Metrics. We employ two metrics to evaluate each model’s predicted captions and segmentations, respectively. We do use these after first applying a variant of GLaMM’s Hungarian Matching [21] to align predictions with the ground truth. This alignment considers granularity by matching based on class names, assigning an IoU of 0 to missed entities.

To evaluate *captions*, we use METEOR⁵ [1], which mea-

²Special tokens are excluded from the example template for clarity but are included in the ground truth data.

³The context-free grammar and a proof that the captions form a context-free language are provided in the supplementary materials.

⁴While the GROUNDHOG [65] model is also a good fit, we could not use it since the authors have not released their code at the time of writing.

⁵While newer metrics like LLM-as-a-judge [69] have been shown to better align with human preferences [5], we found them open-source variants (that enable reproducibility) to be inaccurate for our task. More details are provided in the supplementary materials.

sures the harmonic mean between word token precision and recall. Scores range from 0 (incorrect) to 1 (perfect match).

To evaluate *segmentations*, in line with prior semantic segmentation literature [28, 36], we employ mean Intersection over Union (mIoU) to compare each predicted segmentation against the ground truth. Scores range from 0 (no overlap) to 1 (full overlap).

Subpart-Level Models. We benchmark five models for performing hierarchical semantic segmentation to the subpart granularity.

We include HIPIE [57], the only published method for this task with reproducible evaluation code.⁶ We report HIPIE’s zero-shot performance, as it is designed for open-vocabulary hierarchical segmentation and intended to generalize directly to new vocabularies.⁷ The model requires a list of target categories, so we provided the full set of 254 categories from SPIN [36]. We use the Swin-L [33] variant for its stronger representational power over ResNet-50 [17].

We also introduce a baseline, mentioned in Section 1, that *could* streamline the task into a single inference pass by using multiple parallel decoder heads to generate multiple segmentations at differing granularity levels. We call it Multi-Head. We implemented it using a traditional supervised segmentation model based on an ImageNet-21K [8] pre-trained Swin-L [33] backbone, a Feature Pyramid Network (FPN) [31], and three segmentation heads with convolutional layers and ReLU activations. This model serves as another fully supervised comparison to HALLUMI.

Finally, since our implementation extends the GLaMM [43] model, we consider three variants based on it that can generate both segmentations and captions. Two are intended to reveal the model’s performance in a zero-shot setting, when used without task-specific fine-tuning. These are off-the-shelf GLaMM models fine-tuned for specific tasks: (1) GLaMM-GCG is trained exclusively for Grounded Conversation Generation (GCG) tasks, which entails producing a comprehensive description of a region with interleaved segmentation masks and (2) GLaMM-All is trained on multiple vision-language tasks, including GCG, referring segmentation, region captioning, and visual question answering. The third variant, which we call GLaMM-Oracle, is a specialized version fine-tuned for SPIN instructed to localize each entity, one inference pass at a time. Although it simulates an unrealistic, idealized scenario with prior knowledge of entities, it offers insight into a potential upper-bound of performance. This variant outputs segmentations only, following the SPIN setup [36].

⁶Two other models, HSSN [28] and ViRReq [49], also can support subpart-level hierarchical segmentation, but their code was either unavailable or led to results that differed from the original publication, and their subpart-level performance was not evaluated on the SPIN dataset [36].

⁷Notably, fine-tuning HIPIE was not an option because its codebase lacks fine-tuning hooks and embeds dataset-specific assumptions.

Method	SPIN [36]			
	mIoU _{Obj}	mIoU _{Part}	mIoU _{Sub}	METEOR
HIPIE [57]	21.69	8.23	0.92	-
GLaMM-GCG [43]	26.69	2.48	3.98	0.02
GLaMM-All [43]	27.85	10.37	4.25	0.01
Multi-Head	74.05	37.61	6.89	-
PartGLEE [27]	83.38	60.93	-	-
HALLUMI	89.30	58.15	18.45	0.48
GLaMM-Oracle [36]	91.08	60.76	24.56	-

Table 1. Performance of HALLUMI and six hierarchical segmentation models. HALLUMI outperforms all baselines across most granularities and approaches the oracle upper bound [36]. Models above the line are zero-shot, while those below are fine-tuned. METEOR scores are only reported for variants that generate captions and segmentations: HALLUMI and GLaMM variants.

Part-Level Model. We also evaluate a state-of-the-art foundation model designed for hierarchical object-part semantic segmentation: PartGLEE [27]. It uses a modified Q-Former [26] to decompose object queries into part queries that are processed by a separate decoder. It is designed for instance segmentation, so we convert its predictions to semantic segmentations by merging all predictions of a given class if a prediction score exceeds a threshold of 0.5. This threshold balances part discovery while mitigating unconfident false positives. We utilize the Swin-L variant of PartGLEE trained on various hierarchical object-part semantic segmentation datasets, including PartImageNet [15] and PACO [42]. SPIN shares the same part and object annotations as PartImageNet, enhancing compatibility.

4.2. Hierarchical Segmentation Results.

Overall Performance. Results are shown in Table 1.

Across all granularity levels, our method either outperforms or has comparable performance to all baselines (excluding the oracle method designed as an upper bound). Notable gains are especially observed at finer granularities. For instance, HALLUMI achieves a 6 percentage point (pp) increase in mIoU for *object segmentation* over the next best performing method, PartGLEE, and a 11.66pp gain at the subpart level over the next best performing method, Multi-Head. While our method performs better overall, in some cases it performs worse for parts; e.g., trails PartGLEE by 2.78pp. We discuss this limitation more below. When compared to the upper bound, GLaMM-Oracle, which reflects the rare case where users have full access to all categories they wish to segment, our approach achieves competitive performance at all granularities, with the biggest gap at the subpart level, which we attribute to missed entities, an affordance not lost upon the oracle setting.

A commonality across all methods including ours is that performance *decreases* as granularity *increases* (e.g., moving from objects to subparts). We attribute this to two key

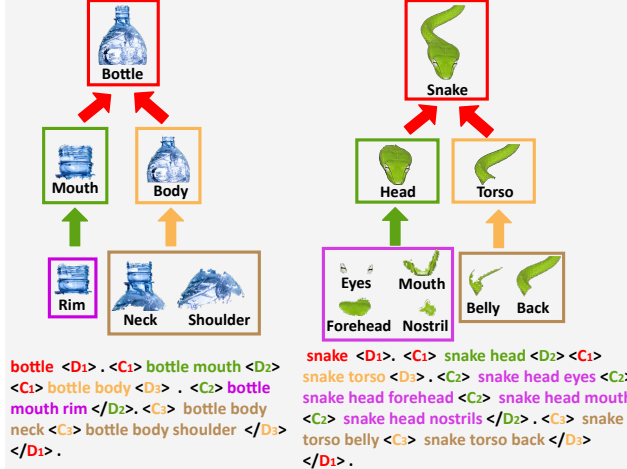


Figure 3. A short model-produced caption with few entities (left) and a long model produced caption with more entities (right). Overall, we do not observe a difference in segmentation quality despite the difference in caption lengths. We omit all special tokens (<p>, </p>, and <SEG>) aside from our proposed hierarchical tokens for readability.

factors. First, as discussed in [36], regions at finer granularities are smaller than their parent regions. This reduction in size makes them more sensitive to pixel-level misclassifications, exacerbating the impact of these errors when computing IoU. Second, most of the models we evaluated, except for the benchmarked Multi-Head model, rely on CLIP [41]. CLIP is used either as the whole image encoder (e.g., in GLaMM-based models) or as the text encoder (e.g., in HIPIE). However, CLIP has been shown to struggle with fine-grained entity recognition [59], leading to suboptimal representations for sub-object-level regions. This reliance suggests that existing model architectures may not be fully suited to handle hierarchical understanding at finer granularities, as they inherit these limitations from CLIP.

Subpart-Specific Performance. Our approach is 2.5-19 times better with respect to subparts compared to all five non-oracle baselines, with all baselines achieving nearly negligible success (i.e., less than 7% mIoU). We hypothesize the considerably worse performance stems from a domain mismatch between categories in the SPIN dataset and those observed by models during training.

For instance, for GLaMM models operating in a zero-shot setting and trained on closely related tasks (i.e., GLaMM-GCG and GLaMM-All), we suspect captions used for training typically describe entities with respect to their shape and other intrinsics [53] (e.g., color, texture) rather than details about an entities hierarchical decomposition. This is supported by the very low METEOR scores, which indicate a struggle to produce similar captions.

Similarly, we observe that HIPIE struggles to recognize

entities outside of its training data. This issue can potentially be attributed to its sensitivity to the hyperparameters that adjusts the importance of CLIP features versus its classification logits when performing zero-shot segmentation. This sensitivity affects HIPIE’s ability to generalize to new settings without extensive tuning. This highlights a downside of HIPIE’s design, as it requires careful adjustment of hyperparameters to adapt to novel scenarios rather than achieving its claim to support open vocabularies as is.

We suspect Multi-Head’s poor subpart performance stems from the limited training signal provided by subpart pixels. The sparsity of these pixels, coupled with the absence of explicitly defined relationships between outputs, constrains the model’s ability to learn detailed hierarchical structures. This suggests that explicit design choices for relational matching are essential for hierarchical performance, as evidenced by HALLUMI’s superior results.

Category-Level Performance. We next compare performance for subparts shared across multiple object classes, such as “eyes” which are shared by quadrupeds and bipeds. We compare the distribution of IoU values for these subparts using the Kruskal-Wallis test [20]. Specifically, for each subpart grouping with k ($k \geq 2$) IoU distributions X_i , where each X_i represents one object-part-subpart class, we test the null hypothesis H_0 that all X_i share the same distribution. The alternative hypothesis H_1 is that at least one distribution differs. To account for multiple comparisons, we set a significance level of $\alpha = 0.05$ and apply a Bonferroni correction.

Out of 38 higher-order subpart groupings, 14 groups (36%) exhibit p-values below the adjusted α , indicating that at least one IoU distribution within these groupings differs substantially. This result shows that certain shared subparts display varying levels of accuracy across classes, suggesting further improvements could focus on how knowledge of nested parts generalizes to rarer, novel objects. For example, deer and the less common saola share nearly identical parts and subparts (such as four legs, a head, eyes, antlers). Understanding these overlaps may lead to better segmentation for more underrepresented categories.

4.3. Model Design Analysis

We next conducted ablation studies to gauge the importance of our design choices for the HALLUMI framework.

Does Human-Sounding Language Help Hierarchical Grounding?

We tested whether adding compositional phrases improves performance in our hierarchical captioning task. We used a succinct format with reduced natural language flow, employing breadth-first descriptions such as: The <image> contains a [object 1]. The [object 1] contains [object 1] [part 1] and [object 1] [part 2] ... The

[object 1] [part 1] contains [object 1]
[part 1] [subpart 1]

Our results showed a 1.22 percentage point drop in subpart segmentation performance when including relational descriptors like *contains*, suggesting that large language models may not benefit from the extra context in our task. The largest effect was on latency. Longer caption sequences moderately increased inference times. Captions from Section 3 averaged 102 tokens, taking about 26 seconds, while captions with added context averaged 128 tokens, taking 31 seconds. Omitting special tokens lowers inference to about 13 seconds at the expense of some performance and interpretability, illustrating the flexibility of our framework for real-world use where speed may be prioritized. While HALLUMI’s main computational bottleneck, autoregressive decoding, could achieve up to a five-fold speedup if paired with optimized inference frameworks like vLLM [23] (depending on hardware and batch size), our findings underscore the extra time-consuming context is also not desirable from a quality perspective.

Impact of Sequence Length. To analyze the relationship between sequence length and segmentation performance, we performed a linear regression on the length of the predicted caption for an image and the mIoU for each granularity within that image. Formally, we fit a linear regression model, $\hat{y}_i^g \sim \hat{\beta}_1 L_i + \hat{\beta}_0$, where $\hat{y}_i^g$ is the mIoU for granularity $g \in \{\text{object, part, subpart}\}$ for the i^{th} sample of the test set, and L_i is the length of the predicted caption.

Overall, we observe that the $\hat{\beta}$ parameters are statistically significant (using a significance threshold of $\alpha = 0.001$), but the correlations between sequence length and segmentation performance are weak, with R^2 (Pearson correlation coefficient) values of 0.01, 0.08, and 0.18 for objects, parts, and subparts, respectively. On average, while the length of the caption produced by the segmentation model has some effect on the model’s IoU performance, it does not sufficiently explain the variability in performance.

Qualitative results of short captions produced by our model alongside longer captions in Figure 3. Overall, this exemplifies the minor differences in segmentation quality and entity recognition, despite the varied caption lengths.

How is the Autoregressive Conditioning Helping? We analyzed how autoregressive modeling is benefiting our task. The main benefit can be attributed to *how* we construct the hierarchy. Since causal attention conditions tokens only on the preceding tokens, we discourage repeated entities (e.g., only 0.004% of entities are repeated during inference on SPIN’s test set) and allow prior, coarser-grained entities to inform the model about which categories may come next (e.g., by learning co-occurrences in a hierarchy path). However, due to the underlying architecture of our base

% of Training Data	mIoU _{Obj}	mIoU _{Part}	mIoU _{Sub}	METEOR	Recall
1%	0.00	0.00	0.00	0.04	0.00
10%	87.70	52.88	13.44	0.32	92.00
25%	88.19	54.35	15.59	0.32	91.57
50%	88.78	55.42	15.12	0.32	90.75
100%	89.30	58.15	18.45	0.48	94.09

Table 2. Ablation study on amount of training data. We observe stable object performance, while parts and subparts nearly always improve with more data.

model, GLaMM, each segmentation mask is produced *independently* of other entity masks because we apply causal attention solely on the *text* outputs and not on the segmentation masks. This means that while the text generation benefits from prior context, the segmentation masks do not with our current architecture, as they are produced without considering other masks. This suggests that although vanilla causal attention aids in entity *recognition* through the sequential text, more sophisticated attention mechanisms may be needed to effectively incorporate prior mask information into the segmentation process.

Impact of Amount of Training Data. Next, we analyze how sensitive our approach is to operating in a low-data regime. To this end, we created random splits comprising 1%, 10%, 25%, and 50% of the SPIN training set. Results are shown in Table 2, where we additionally include recall across *all* granularities to assess how training data influences HALLUMI’s ability to correctly discover entities.

Using only 1% of SPIN’s training data (88 images), GLaMM fails to learn our hierarchical captions, reflected by its near-zero METEOR score, which hinders hierarchical segmentations. With 10% of the data, GLaMM begins to learn our captions and improves performance. Beyond 10%, object-level performance plateaus (e.g., a 0.49pp gain from 10% to 25%), with consistent increases in part performance. When moving from 25% to 50%, object and part performance rise by 0.67pp and 1.07pp, but subpart performance drops by 0.47pp. We attribute this to the models learning at multiple granularities: more data boosts object and part segmentation yet increases variability for subparts. Nevertheless, we observe high recall (10%-100% of data), indicating effectiveness at identifying correct entities with relatively little supervision. Subparts gain the most from using the full dataset, showing a 22% relative improvement compared to using only half of the available data. This gap likely arises because the pre-training data used by GLaMM, Grand [43], is largely object-centric, offering few subpart examples and focusing on scene descriptions rather than hierarchical structures of single objects.

Impact of Parent Context. There are many ways to refer to parts and subparts when constructing captions. In HALLUMI, we prepend higher-level entities when we refer

Context	mIoU _{Object}	mIoU _{Part}	mIoU _{Subpart}
✗	53.14	25.66	0.01
✓	89.30	58.15	18.45

Table 3. Ablation study on referring to finer-grained entities. We either refer to an entity at its current granularity (e.g., eye) or prepend higher-level context (e.g., quadruped head eyes). Overall, we observe large performance gains when including this context.

Tokens	mIoU _{Object}	mIoU _{Part}	mIoU _{Subpart}	Recall
✗	89.30	57.41	15.81	90.75
✓	89.30 (+0)	58.15 (+0.74)	18.45 (+2.64)	94.09 (+3.34)

Table 4. Ablation study for including hierarchical tokens in model predictions vs. not including. We observe comparable object performance with large gains at the subpart granularity.

to entities at the current granularity (e.g., “quadruped head eyes”). We find that this approach outperforms the naive method of referring only to entities at the current granularity (e.g., “eyes”). Results are shown in **Table 3**.

Overall, we observe a significant performance difference between the two referring strategies. Most notably, when referring to an entity solely by its current granularity, we are nearly unable to properly segment any subparts (i.e., achieving near 0 mIoU for subparts). In contrast, including higher-level context improves segmentation performance substantially. This observation reinforces findings from both OV-Parts [59] and SPIN [36] that foundation models lack hierarchical understanding, especially as granularity increases, attributed to inherent model limitations or lack of hierarchical representation in training data.

Impact of Hierarchical Tokens. We next examine the effect of including our hierarchical tokens from Section 3, as shown in **Table 4**. Overall, object- and part-level performance remains comparable, while subpart-level performance increases by 16.7% when these tokens are present. We hypothesize that their explicit relational cues help the model contextualize finer-grained entities. Additionally, including our tokens increases recall for hierarchical content by 3.34 percentage points. This improvement suggests that signaling when to stop or continue decomposing an entity helps reduce false positives.

Even without any performance gain, these tokens prove useful for explicitly recovering hierarchical relationship trees, since the model directly classifies parent-child connections (**Figure 2**). This approach avoids heuristic post-processing when tokens are absent, while also enhancing interpretability by explicitly communicating how entities relate to each other.

Impact of Hierarchy Depth. We analyzed the effects of limiting hierarchical depth to parent-child relationships on segmentation performance by training HALLUMI on

SPIN with only object and subpart annotations. While this approach had minor object-level performance differences (0.21 percentage points), part-level performance dropped by 4.86 percentage points. We hypothesize that this decrease is due to the critical role played by non-terminal nodes in guiding the learning process, which provides essential, explicit context for learning accurate representations. Without these signals, the model struggles to capture detailed part-level information, reducing spatial precision and overall performance. This result highlights recent findings [4] that visual tokens (i.e., image patches) capture more fine-grained information at the sub-part level, indicating that they can reduce uncertainty about part labels effectively, and also shows a benefit of using full hierarchical context for accurate multi-level segmentation.

5. Conclusion

We introduced HALLUMI, an autoregressive framework that recasts hierarchical semantic segmentation as Grounded Conversation Generation [43]. This approach generates hierarchies in a single inference pass, without extra mask decoders. By using specialized tokens for hierarchical decomposition, HALLUMI explicitly encodes parent-child relationships, enabling efficient part-whole recognition and localization. We demonstrated competitive segmentation results on SPIN [36] compared to contemporary methods. Our experiments show that architectures like GLaMM can learn hierarchical recognition, though performance varies across granularities. Incorporating hierarchical context into generated captions further boosts model performance.

A valuable direction for future work is to improve the computational efficiency of autoregressive models for hierarchical segmentation. Parallel decoding strategies such as superposed decoding [46] and optimized inference frameworks like vLLM [23] offer promising paths to reduce the latency introduced by autoregressive modeling. Replacing GLaMMs dual-encoder architecture with a more streamlined design could further cut inference costs while maintaining performance.

Another future direction is to improve model performance for datasets showing multiple objects. This effort includes establishing benchmark datasets that support subpart hierarchical analysis, as SPIN (the only subpart semantic segmentation dataset) is limited to single-object images.

Acknowledgments. Josh Myers-Dean is supported by a NSF GRFP fellowship (#1917573). Experiments were conducted on Adobe Research compute resources.

References

- [1] Satantjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 4
- [2] Lorenzo Bianchi, Fabio Carrara, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. The devil is in the fine-grained details: Evaluating open-vocabulary object detectors for fine-grained understanding. *arXiv preprint arXiv:2311.17518*, 2023. 1
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2
- [4] David M Chan, Rodolfo Corona, Joonyong Park, Cheol Jun Cho, Yutong Bai, and Trevor Darrell. Analyzing the language of visual tokens. *arXiv preprint arXiv:2411.05001*, 2024. 8
- [5] Chongyan Chen, Mengchen Liu, Noel Codella, Yunsheng Li, Lu Yuan, and Danna Gurari. Fully authentic visual question answering dataset from online communities. *arXiv preprint arXiv:2311.15562*, 2023. 4
- [6] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2021. 2
- [7] Daan de Geus and Gijs Dubbelman. Task-aligned part-aware panoptic segmentation through joint object-part representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3174–3183, 2024. 2
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4
- [10] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008. 2
- [11] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2241–2248. Ieee, 2010. 2
- [12] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2
- [13] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Xingqian Xu, Trevor Darrell, Nicu Sebe, Zhangyang Wang, and Humphrey Shi. Pair diffusion: A comprehensive multimodal object-level image editor. 1
- [14] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):59–73, 2008. 2
- [15] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 4, 5
- [16] Ju He, Jieneng Chen, Ming-Xian Lin, Qihang Yu, and Alan L Yuille. Compositor: Bottom-up clustering and compositing for robust part and object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11259–11268, 2023. 2
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 4
- [19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Piotr Dollr, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 4
- [20] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952. 6
- [21] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4
- [22] Tobias Kuhn. A survey and classification of controlled natural languages. *Computational linguistics*, 40(1):121–170, 2014. 4
- [23] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 7, 8
- [24] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2, 3
- [25] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 2
- [26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5

- [27] Junyi Li, Junfeng Wu, Weizhi Zhao, Song Bai, and Xiang Bai. Partglee: A foundation model for recognizing and parsing any objects. *arXiv preprint arXiv:2407.16696*, 2024. 5
- [28] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1246–1257, 2022. 1, 2, 5
- [29] Zhiheng Li, Wenxuan Bao, Jiayang Zheng, and Chenliang Xu. Deep grouping model for unified perceptual parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4053–4063, 2020. 2
- [30] Xiaodan Liang, Hongfei Zhou, and Eric Xing. Dynamic-structured semantic propagation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 752–761, 2018. 2
- [31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5
- [32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3
- [33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 5
- [34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 4
- [35] Rei Miyata, Anthony Hartley, Cécile Paris, Midori Tatsumi, and Kyo Kageura. Japanese controlled language rules to improve machine translatability of municipal documents. In *Proceedings of Machine Translation Summit XV: Papers*, 2015. 4
- [36] Josh Myers-Dean, Jarek Reynolds, Brian Price, Yifei Fan, and Danna Gurari. Spin: Hierarchical segmentation with subpart granularity innatural images. In *Computer Vision – ECCV 2024*, pages 275–292, Cham, 2025. Springer Nature Switzerland. 4, 5, 6, 8
- [37] Sarah E Petersen and Mari Ostendorf. Text simplification for language learners: a corpus analysis. In *SLaTE*, pages 69–72, 2007. 4
- [38] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 2
- [39] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European conference on computer vision*, pages 647–664. Springer, 2020. 2
- [40] Lu Qi, Jason Kuen, Weidong Guo, Jiuxiang Gu, Zhe Lin, Bo Du, Yu Xu, and Ming-Hsuan Yang. Aims: All-inclusive multi-level segmentation for anything. *Advances in Neural Information Processing Systems*, 36, 2024. 1
- [41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 6
- [42] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023. 5
- [43] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*, 2023. 2, 3, 4, 5, 7, 8
- [44] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020. 4
- [45] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kun-chang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 2
- [46] Ethan Shen, Alan Fan, Sarah M Pratt, Jae Sung Park, Matthew Wallingford, Sham M Kakade, Ari Holtzman, Ranjay Krishna, Ali Farhadi, and Aditya Kusupati. Superposed decoding: Multiple generations from a single autoregressive inference pass. 2024. 8
- [47] Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024. 4
- [48] Erik B Sudderth, Antonio Torralba, William T Freeman, and Alan S Willsky. Learning hierarchical models of scenes, objects, and parts. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, pages 1331–1338. IEEE, 2005. 2
- [49] Chufeng Tang, Lingxi Xie, Xiaopeng Zhang, Xiaolin Hu, and Qi Tian. Visual recognition by request. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15265–15274, 2023. 1, 2, 5
- [50] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024. 1
- [51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aure-

- lien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 4
- [52] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63:113–140, 2005. 2
- [53] BARBARA Tversky. On exploring parts and wholes. *Computational and cognitive models of creative design VI*, pages 3–16, 2005. 6
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2
- [55] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [56] Wenxuan Wang, Tongtian Yue, Yisi Zhang, Longteng Guo, Xingjian He, Xinlong Wang, and Jing Liu. Unveiling parts beyond objects: Towards finer-granularity referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12998–13008, 2024. 1
- [57] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 5
- [58] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3
- [59] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 6, 8
- [60] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See say and segment: Teaching lms to overcome false premises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13459–13469, 2024. 2
- [61] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 2
- [62] Jingru Yi, Burak Uzkent, Oana Ignat, Zili Li, Amanmeet Garg, Xiang Yu, and Linda Liu. Augment the pairs: Semantics-preserving image-caption pair augmentation for grounding-based vision and language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5520–5530, 2024.
- [63] Chang Yu, Xiangyu Zhu, Xiaomei Zhang, Zidu Wang, Zhaoxiang Zhang, and Zhen Lei. Hp-capsule: Unsupervised face part discovery by hierarchical parsing capsule network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4032–4041, 2022. 2
- [64] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022. 2
- [65] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozhi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *Conference on Computer Vision and Pattern Recognition 2024*, 2024. 2, 4
- [66] Yibiao Zhao and Song-Chun Zhu. Image parsing with stochastic scene grammar. *Advances in Neural Information Processing Systems*, 24, 2011. 2
- [67] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3
- [68] Long Zhu, Yuanhao Chen, Chenxi Lin, and Alan Yuille. Max margin learning of hierarchical configurational deformable templates (hcdts) for efficient object parsing and pose estimation. *International journal of computer vision*, 93:1–21, 2011. 2
- [69] Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023. 4