Think Before You Drive: World Model-Inspired Multimodal Grounding (Supplementary Materials)

1. DrivePilot Dataset

1.1. Step-1: RAG In-Context Learning

To enhance LLM reasoning with real-world driving knowledge, we implement a two-tier Retrieval-Augmented Generation (RAG) framework. This process grounds AV visual grounding annotations in empirical driving data, ensuring contextual accuracy and reduced hallucination rates. We curate a multimodal knowledge base, including comprehensive expert-annotated scenarios from the nuScene dataset, covering agent trajectories, road topology, and traffic rule compliance. For each query pair (input image and command), we execute a three-phase retrieval process:

Feature Encoding. A pre-trained vision backbone (Fast R-CNN) extracts dense visual embeddings from the input image, capturing spatial relationships and object semantics. Simultaneously, a BERT-based language model encodes textual features from the given command, ensuring a rich semantic representation for cross-modal alignment.

Cross-Modal Retrieval. The extracted embeddings are used to retrieve the top-k ($k = 5$) most relevant scenes from the knowledge base via cosine similarity. The retrieved samples include both human-annotated metadata and raw sensor data from the traffic reports and nuScene dataset, enhancing scene understanding and context recall.

Knowledge Infusion. The retrieved scenarios are formatted into a structured prompt that guides the LLM to align its reasoning with historical driving patterns, such as how vehicles yield to pedestrians in the rain, and legal precedents, such as how to interpret ambiguous traffic lights. This ensures that the LLM's decisions are contextually based, using real-world driving behaviors and legal standards to improve situational awareness and reasoning accuracy.

1.2. Step-2: CoT Annotation Generation

LLMs like Qwen and LLaVA excel in natural language understanding but are not inherently trained for AD or VG tasks. Prior studies [3, 6, 8] have shown that structured prompt engineering can significantly improve LLMs' zero-shot visual description performance. To leverage this potential, we develop a progressive Chain-of-Thought (CoT) prompting strategy for generating context-aware semantic annotations, enabling dataset augmentation and semantic enrichment without fine-tuning. Specifically, we explore the utility of Qwen in augmenting and refining existing multimodal data (frontal and BEV images, paired with natural language commands) using few-shot or zero-shot prompting techniques. This enables dataset expansion and improved semantic annotation generation without costly and time-consuming fine-tuning. This process includes:

Multiview Image Annotation. As illustrated in Table 1, we fuse frontal-view imagery with bird's-eye-view (BEV) spatial data to generate rich scene annotations. For each object detected in the frontal view (vehicles, pedestrians), we compute its BEV coordinates using a calibrated projection matrix derived from the camera's intrinsic parameters and lidar-camera extrinsic calibration. This dual-view align-

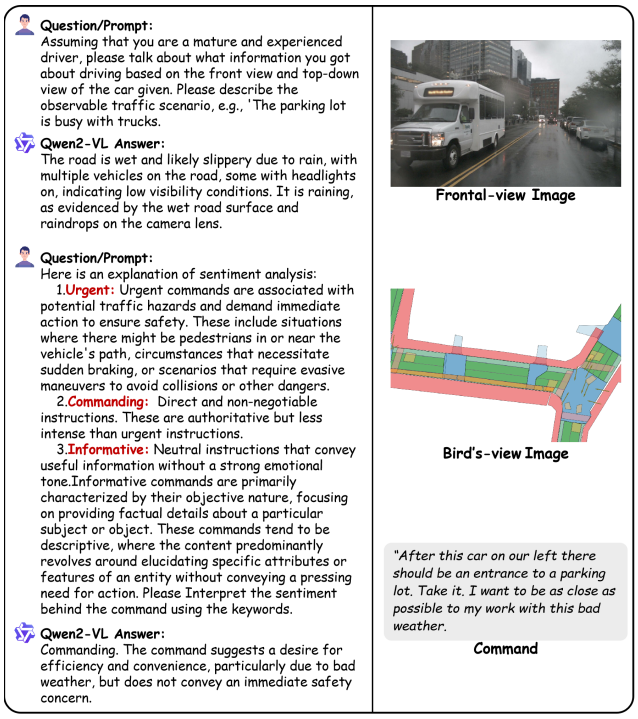


Figure 1. An example of the Qwen2-VL leveraging lens-less cueing technology to interpret driving scenarios and generate driving maneuvers, eliminating the need for specialized fine-tuning.

ment enables Qwen to reason about both object appearance and spatial context. The annotation process is as follows:

Scene Semantic Enhancement. Beyond object-level annotations, we introduce Scene Semantic Enhancement to enrich high-level contextual information. As shown in Table 1 and Figure 1, we employ CoT prompting, guiding Qwen to generate 14 categories of contextual metadata, including:

- Scene Descriptions: Summarizing overall environmental context, including road surface conditions, weather, lighting, and potential hazards in the traffic scenes.
- Emotion Interpretation: Analyzing pedestrian and driver intent to predict potential interaction risks.
- Road Condition Summaries: Detailing surface quality, lane markings, and obstacles affecting navigation.
- Traffic Signal & Sign Interpretations: Detailing surface quality, lane markings, and obstacles.

To diversify training data, we randomly replace 30% of the original command texts with this CoT-generated augmentation text during training, ensuring diverse linguistic patterns and robust generalization. Furthermore, keyword-based augmentation is introduced to append relevant context cues to the original command prompts, aiding semantic disambiguation. For example, semantic keywords such as "low visibility" and "intersection" are added as auxiliary hints to commands in scenarios with obstructed vision or Multi-agent. Notably, this step-by-step process involves dialogues where each "thought" guides Qwen2-VL to understand different aspects of the scene or command. The first thought focuses on understanding the scene and identifying key objects and their dynamics. The second thought analyzes command keywords and emotions. After h iterations of reasoning and updating (with iterations varying per sample for the actual situation), insights from each thought are synthesized into a cohesive semantic annotation, uniformly formatted for model processing.

1.3. Step-3: Manual Cross-check Validation

To ensure high fidelity and compliance, all LLM-generated annotations undergo a rigorous multi-stage review by 13 domain experts, including AV safety engineers and certified instructors. Annotations are validated against nuScenes multimodal sensor data (LiDAR, radar, and camera) to ensure spatial precision. Specifically, object positions and spatial relationships are corroborated with 3D ground-truth coordinates, while temporal consistency is verified across frames to align with actual motion trajectories. Furthermore, BEV annotations are manually audited to resolve spatial ambiguities and ensure precise depth perception, effectively eliminating false positives.

2. Spatial-Aware World Model

The calculation of depth-derived prior $P(k)$ is derived from the depth graph F_d, which encodes the depth information of the input visual data. To ensure a consistent depth representation, F_d is normalized to the range $[0, 1]$ using an Exponential Decay Function. This function assigns higher values to closer objects while attenuating the influence of distant regions, ensuring depth-aware feature refinement. The transformation is formally expressed as follows:

$$F_D^{\text{nor}}(x) = \exp(-\alpha \cdot F_D(x)) \tag{1}$$

where α is a decay rate hyperparameter that regulates depth sensitivity, preserving finer details for nearby objects while suppressing distant regions. Pixels corresponding to objects at infinite depth are set to zero, excluding them from further processing to improve computational efficiency.

Next, the normalized depth map F_D^{nor} is passed through a Multi-Layer Perceptron (MLP), which employs a piecewise activation function. This allows the model to adaptively emphasize depth regions based on their visual importance, refining spatial awareness in visual grounding. The transformation is defined as follows:

$$P(x) = \phi_{\text{MLP}}(F_D^{\text{nor}}(x)) \tag{2}$$

where ϕ_{MLP} represents the MLP transformation, mapping depth-normalized features for downstream tasks.

3. Training Loss

To train our model, we define supervision over the prediction tuple $\{s, y\}$ and the ground-truth tuple $\{\hat{s}, \hat{y}\}$, where Z_v denotes the visual latent states produced by SA-WM. For each visual block, the model is required to regress the IoU between the predicted region and the annotated ground-truth region, which is denoted as y. This design ensures that every visual block contributes explicit grounding signals rather than relying solely on a final aggregated score.

Stage-1: World-Model Rollout Pretraining. In the first stage, the supervision is dominated by dense, patch-level observations in complex traffic scenes. Under such settings, standard region-based losses like Dice loss or vanilla cross-entropy frequently suffer from severe foreground–background imbalance, caused by large fields of view, multiple reference objects, and a relatively small proportion of positive pixels. To mitigate this, we draw inspiration from lesion detection in medical imaging and combine a Tversky loss $\mathcal{L}_{\text{tve}}$ [9] with Focal loss $\mathcal{L}_{\text{foc}}$ [7]. During World-Model rollouts, we initialize the prior score S with uniform probability to accelerate the convergence of visual-text mapping. By leveraging bilinear interpolation on the downsampled ground-truth mask $\hat{S}$, we enforce a dense alignment constraint. Unlike standard formulations that simply sum losses, we construct a weighted integration of a re-parameterized Tversky objective and a hardness-weighted Focal penalty. Let $p_{ij} \in S$ and $g_{ij} \in \hat{S}$ be the predicted probability and binary ground truth at location (i, j).

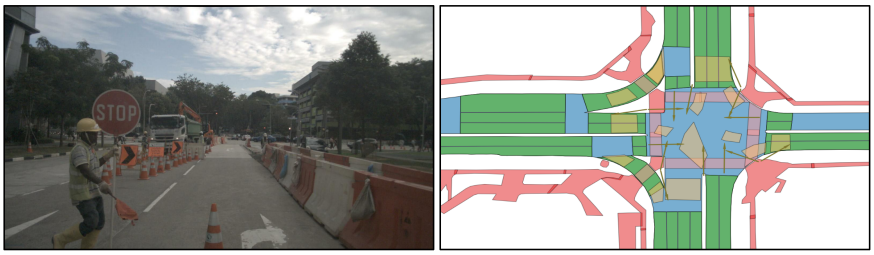
Image Annotations	
Visual input	 Figure 2. An example of how a large language model interprets visual information: the left image represents the front view, while the right image corresponds to the BEV image.
Prompt	You are an AI visual assistant tasked with providing semantic enhancement for visual grounding tasks. Given an image containing multiple bounding boxes and its corresponding BEV (Bird's-Eye View) map, your role is to annotate each region by supplementing and describing the objects within it. In the BEV map, green areas represent drivable regions, while red areas indicate non-drivable regions. You should concisely summarize the behavior of objects in each region and their interactions with the surrounding environment, then describe the specific content of the provided bounding boxes, including object types, actions, positions, and relative locations between objects. Below are the coordinates for each entity object in the forward view and BEV map, along with the corresponding annotation. Example: • **Entities Annotation 1:** (label 1: [x1,y1,x2,y2]) Description of the features in this scene region. • **BEV Annotation 1:** (label 1: [x1,y1,x2,y2]) Description of the spatial positioning in this region.
Entities Bounding box	Person:[0.116,0.45,0.314,0.998], Truck[0.322,0.403,0.488,0.609], Sign: [0.202,0.338,0.329,0.553], Car [0.685,0.541,0.756,0.630], etc.
BEV coordinates	Entities BEV coordinates, Drivable area coordinates, Non-drivable area coordinates.
LLM Annotations	**Entities Annotation 1:** (Person: [0.116,0.45,0.314,0.998]) A worker in safety gear is holding a stop sign, managing traffic at a road construction site. **Entities Annotation 2:** (Truck: [0.202,0.338,0.329,0.553]) A truck is involved in construction activities or waiting to proceed, on the left side of the road ahead. **Entities Annotation 3:** (Stop Sign: [0.202,0.338,0.329,0.553]) A warning sign being held by a worker crossing the road. **Entities Annotation 4:** (Car: [0.685,0.541,0.756,0.630]) A black sedan approaching head-on in the right lane. ... **BEV Annotation 4:** (Car: [0.685,0.541,0.756,0.630]) A sedan positioned at the intersection behind a truck.

Table 1. Illustration of LLM-driven scene understanding. Given paired front-view images and BEV maps, the LLM is prompted with region definitions, bounding boxes, and BEV coordinates to produce rich semantic annotations. The prompt instructs the model to summarize scene context, describe object behaviors, and provide detailed entity-level grounding (types, actions, positions, and relations). By integrating 2D visual data with 3D spatial coordinates (BEV), the model generates fine-grained semantic descriptions. This demonstrates the LLM's ability to reason about object behaviors and interactions within a dynamic traffic environment.

We define the intersection and set differences for class k as:

$$\mathcal{I}_k = \sum\nolimits_{i,j} p_{ij} g_{ij},$$
$$\mathcal{F}_k^{\alpha} = \alpha \sum\nolimits_{i,j} (1 - g_{ij}) p_{ij}, \quad \mathcal{F}_k^{\beta} = \beta \sum\nolimits_{i,j} (1 - p_{ij}) g_{ij} \tag{3}$$

where ϵ is a smoothing factor, and α, β control the penalty magnitude for False Positives (FP) and False Negatives (FN), respectively, allowing the model to dynamically shift focus during training. The rollout loss, $\mathcal{L}_{rol}$, is defined as:

$$L_{\text{roll}} = \lambda_{tve} \sum_{k=1}^{K} \left[1 - \frac{\mathcal{I}_k + \epsilon}{\mathcal{I}_k + \mathcal{F}_k^{\alpha} + \mathcal{F}_k^{\beta} + \epsilon} \right] - \lambda_{\text{foc}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{H}(p_n, g_n) \tag{4}$$

Here, the λ_{tve} and λ_{foc} are both the hyperparameters. Moreover, $\mathcal{H}(\cdot)$ represents the focal modulation defined as $\mathcal{H}(p_n, g_n) = \alpha_t (1 - p_{t,n})^{\gamma} \log(p_{t,n})$, where $p_{t,n}$ reflects the model's confidence in the true class. This formulation ensures that gradients are dominated by hard-to-classify examples like small distant vehicles, rather than the dominant background, effectively regularizing the rollout trajectory.

Stage-2: Grounding Decision Supervision. Upon establishing robust feature representations, the second stage focuses on precise localization. To prevent catastrophic forgetting of the patch-level priors learned in the first stage, we employ a multi-task learning strategy. We define the grounding loss L_{ground} as a hybrid constraint optimization that simultaneously minimizes the distributional divergence and the geometric regression error. Let $\hat{\mathbf{y}}$ be the ground truth bounding box and S_{lat} be the latent state output representing the object mask. The objective function aggregates the Binary Cross-Entropy (BCE) and L1 regression error, normalized over the batch. The loss L_{gro} is defined as:

$$L_{\text{gro}} = \mathbb{E}_{(y,S)\sim\mathcal{D}} \left[\lambda_{\text{cls}} \cdot \Psi_{\text{BCE}}(y, \hat{y}) + \lambda_{\text{reg}} \cdot \| S_{lat} - \hat{S} \|_1 \right] \tag{5}$$

where Ψ_{BCE} is the binary cross-entropy operator, and $\| \cdot \|_1$ imposes a sparsity-inducing L1 penalty on the mask prediction. The hyperparameters λ_{cls} and λ_{reg} balance the trade-off between semantic classification accuracy and geometric precision. Overall, this diversity loss term incentivizes the model to capture cross-modal interactions and produce target predictions consistent with the commander's intent.

4. Experiments Setups

4.1. Benchmarks

To evaluate our model's effectiveness, we conduct experiments on the dataset zoo: Talk2Car, DrivePilot, MoCAD, and RefCOCO, RefCOCO+, and RefCOCOg. These datasets provide diverse and complex real-world scenarios for benchmarking visual grounding in autonomous driving.

Talk2Car. The Talk2Car dataset [2], an extension of the NuScenes dataset [1], consists of 11,959 natural language commands across 9,217 images captured in urban landscapes of Singapore and Boston. This dataset includes a variety of conditions, such as different times of day and weather scenarios, offering a challenging and diverse benchmark. The commands, averaging 11 words, contain complex instructions (e.g., "Parallel park behind the black car on our right"), requiring precise semantic understanding and scene reasoning. It enhances the NuScenes with bounding box annotations across 850 videos, with 55.94% of commands originating from Boston and 44.06% from Singapore. A detailed linguistic analysis reveals an average of 11.01 words per command, comprising 2.32 nouns, 2.29 verbs, and 0.62 adjectives, highlighting the linguistic diversity and complexity of the dataset. Each video is associated with an average of 14.07 commands, enriching the contextual learning process. The dataset is split into training (8,349 commands, 69.8%), validation (1,163 commands, 9.7%), and test sets (2,447 commands, 20.4%).

DrivePilot. We introduce DrivePilot, the first dataset to leverage Qwen's linguistic capabilities for detailed semantic scene annotation using regularized prompts. This dataset categorizes urban scenes across 14 dimensions, including weather conditions, emotional context, and agent interactions. Each dataset entry comprises a natural language command, paired front-view and BEV images, scene annotations generated by Qwen2-VL, and precise target object locations. The dataset is designed to challenge models with object disambiguation and complex query interpretation, closely reflecting real-world AV navigation challenges.

MoCAD. The MoCAD dataset originates from the first Level 4 autonomous bus deployed in Macau and has been continuously tested since 2020. The dataset spans 300+ hours of real-world driving, including data sets from a 5-kilometer campus route, a more extensive 25-kilometer city and urban road collection, and various open traffic situations observed under varying weather, time, and traffic density conditions. It comprises over 13,000 scene images and nearly 40,000 scene objects, with an average command length of 12.5 words, providing a rich dataset for visual grounding research. A distinctive aspect of MoCAD is its Macau-based driving environment, where right-hand driving contrasts with regions that enforce left-hand driving.

RefCOCO. This dataset was collected using the ReferItGame, a two-player interactive game where one player describes an object in an image, and the other identifies it based on the description. It contains 142,209 referring expressions for 50,000 objects across 19,994 images. The dataset is divided into training, validation, and test sets, with test images further categorized into Test A (images containing multiple people) and Test B (images containing multiple objects other than people).

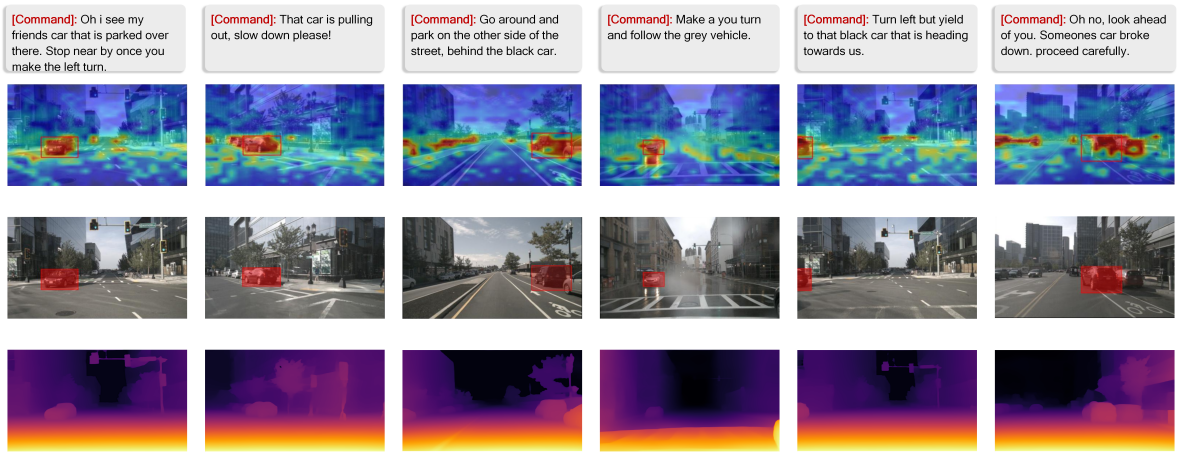


Figure 3. Visualization of ThinkDeeper's multimodal grounding. The first line corresponds to the query commands for each scene, while the second row shows the future latent states Z_v generated by the SA-WM alongside the corresponding prediction. The third row highlights the ground truth regions in red mask areas, while the fourth row presents the depth map, providing spatial context for scene understanding.

RefCOCO+. Similar to RefCOCO, this dataset was also collected through the ReferItGame but with a restriction that prohibits the use of location-based descriptions. This constraint encourages the use of appearance-based expressions, making the task more challenging. RefCOCO+ comprises 141,564 referring expressions for 49,856 objects in 19,992 images. The dataset shares the same split structure as RefCOCO, including the Test A and Test B subdivisions.

RefCOCOg. Collected in a non-interactive setting, RefCOCOg features longer and more complex referring expressions, averaging 8.4 words per expression compared to 3.5 words in RefCOCO. It includes 95,010 expressions for 49,822 objects across 25,799 images. It is split into training, validation, and test sets, focusing on more descriptive and detailed language, which poses additional challenges for language comprehension and visual grounding models.

4.2. Implementation Details

Overall Configuration. Input images are resized to 384×384 pixels, with text expressions truncated to a maximum of 50 tokens. During training, LLM-augmented text replaces original input descriptions with a 30% probability, followed by keyword text appended after a [SEP] token. We implement a learning rate warmup strategy (10% of total training steps) and use the AdamW optimizer with a batch size of 32. All components except the vision-language feature extractors (ViT and BERT) share an initial learning rate of 10^{-4}. ViT and BERT are initialized with pre-trained weights from BLIP [5], while other network components use Xavier initialization [4] for initialization.

Text Encoder. We use a pre-trained BERT model for text embedding extraction, configured with 16 hidden layers and an embedding vocabulary of 30,524 tokens. We also enforce a maximum sentence length of 50 tokens, with a layer normalization epsilon of 1e-12, and a hidden size calibrated to $d = 768$ for linguistic input processing.

Vision Encoder. We use a Vision Transformer-Base (ViT-B) as the vision encoder with a 4:1 MLP to embed dimension ratio and 12-head multi-head attention, extracting a 24×24 visual token stream, and 3 attention layers.

Spatial-Aware World Model. We use a three-layer cross-modal attention layer ($N = 3$) to compute prior scores, where each layer's output is projected through a linear head. Each cross-modal attention block uses a hidden size of $D = 768$ for both text and visual inputs, with 8 attention heads and a dropout rate of 0.1. The learnable parameters in the discriminative module are initialized with $\mu = 1.0$ and $\sigma = 1.0$. Training proceeds in two stages: we first optimize the model with the world-model rollout loss L_{rol} for 15 epochs, followed by grounding-focused training with L_{gro} for an additional 55 epochs (70 epochs total).

Multimodal Decoder. In the cross-modal hypergraph network, each visual node connects to $L = 8$ text nodes to form hyperedges, selected according to an affinity matrix computed by a 1536-dim MLP. Hypergraph attention uses 4 heads with LeakyReLU (negative slope 0.2), a 768-dim hidden layer, and 0.2 dropout. The multi-layer dynamic attention stack consists of 6 attention blocks (each followed by a linear layer), with 12-head multi-head attention, 768-dim hidden size, and 0.2 dropout. During this stage, the ViT and BERT backbones are kept frozen.

4.3. Corner-case and Long-text Test Sets

To assess model robustness under real-world challenging conditions, we curate four specialized test subsets from the DrivePilot and MoCAD datasets. These include restricted

visibility, multi-agent interactions, ambiguous prompts, and long, complex commands. In the multi-agent set, we focus on scenes containing more than 16 targets. The visual constraints set consists of scenarios with impaired visibility caused by nighttime conditions, fog, rain, camera obstructions, or low-resolution images. To evaluate the model's ability to handle linguistic ambiguity, we identify unclear or potentially ambiguous commands, categorizing them as the ambiguous set. Additionally, recognizing that longer commands often introduce irrelevant details or increased complexity, we select commands exceeding 23 words and categorize them into the long-text test set, designed to assess the model's capacity to process intricate instructions.

5. Visualization

As shown in Figure 3, we present more visualization results generated by DrivePilot. These qualitative results showcase the superior performance of our proposed world model-inspired framework in integrating multimodal, real-world commands by jointly leveraging language, spatial, and visual cues, leading to improved multimodal grounding accuracy. These examples highlight our model's ability to achieve robust localization under challenging conditions like high multi-agent and ambiguous driving scenes.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF CVPR*, pages 11621–11631, 2020. 4

[2] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*, 2019. 4

[3] Jessica López Espejel, El Hassane Ettifouri, Mahaman Sanoussi Yahaya Alassan, El Mehdi Chouham, and Walid Dahhane. Gpt-3.5, gpt-4, or bard? evaluating llms reasoning ability in zero-shot setting and performance boosting through prompts. *Natural Language Processing Journal*, 5:100032, 2023. 1

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 5

[5] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 5

[6] Haicheng Liao, Hanlin Kong, Bonan Wang, Chengyue Wang, Wang Ye, Zhengbing He, Chengzhong Xu, and Zhenning Li. Cot-drive: Efficient motion forecasting for autonomous driving with llms and chain-of-thought prompting. *IEEE Transactions on Artificial Intelligence*, 2025. 1

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[8] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 262–271, 2023. 1

[9] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pages 379–387. Springer, 2017. 2