ChatGPT

Vision-Language Models: Foundations, Generalists, Driving, and Robotics

1. Historical VLM Papers (Foundational Works)

CLIP (2021) – *Contrastive Language-Image Pretraining* by OpenAI revolutionized vision-language learning. CLIP uses a dual encoder (image and text) trained on 400M image-text pairs with a contrastive loss. It maps images and captions into a shared embedding space, enabling zero-shot image classification and retrieval without task-specific training [1]. CLIP's impact was huge: it set the **mainstream paradigm** for high-level visual understanding via language supervision [2]. By achieving ImageNet classification with only text prompts, CLIP demonstrated the power of web-scale image-text data for general vision tasks. Its **architecture** (ViT or ResNet for images + Transformer for text) and training objective have become the template for many VLMs. Recent models still often use CLIP-like encoders or contrastive pretraining as a backbone, though extended with new losses and integration with LLMs.

SigLIP (2023) – *Sigmoid Loss for Language-Image Pre-Training* (Google). SigLIP builds on CLIP but replaces the contrastive softmax loss with a **sigmoid-based binary classification loss** [3]. Instead of forcing a single best label per image, SigLIP treats each image-text pair as independent, allowing "none of the above" when a caption doesn't match an image [4] [3]. This seemingly small change yields **better zero-shot recognition** – SigLIP more reliably outputs low scores when no correct label is present, avoiding the false positives CLIP can produce [4]. SigLIP achieved **superior zero-shot ImageNet accuracy to CLIP** [5], and improved performance on multi-label tasks and visual detection. It retains CLIP's ViT and Transformer encoders, so it's backward-compatible with CLIP architectures [3]. By focusing on individual image-text matches, SigLIP improved robustness and fairness, influencing later VLMs to adopt more flexible loss functions. Google's open release of SigLIP provided the community with a stronger vision encoder for use in multimodal models.

LLaVA (2023) – *Large Language-and-Vision Assistant* (Liu et al., NeurIPS 2023) was a landmark in **connecting vision with large language models**. LLaVA takes a pre-trained image encoder (e.g. CLIP's ViT) and a language model (Vicuna) and **joins them with a projection layer**, then fine-tunes on multimodal instruction-following data [6] [7]. Crucially, LLaVA introduced **visual instruction tuning**: using GPT-4 to generate ~158k image-question-answer pairs, which were used to train the model to follow natural language instructions about images [8]. The result was a chatty assistant that can describe images and answer questions about them, essentially an open-source analog of multimodal GPT-4. LLaVA demonstrated impressive "GPT-4 Vision"-like capabilities on unseen images [9]. Its impact is evident: many subsequent VLMs use LLaVA's approach of combining a frozen (or lightly fine-tuned) vision encoder with a powerful LLM, and leveraging synthetic Q&A data for tuning. LLaVA showed that **in-context reasoning** and dialogue about images are achievable with relatively modest model sizes (Vicuna-13B + CLIP ViT) given the right training data. This paved the way for a proliferation of vision-language instruction models and set a baseline for multimodal chat performance.

Impact and Legacy: These foundational works shaped today's VLMs in different ways. CLIP provided the **universal image-text feature space** that many systems still rely on for encoding pictures [10]. SigLIP demonstrated improvements in **training objectives** for better grounding and zero-shot performance, which influenced successors (e.g. OpenCLIP variants, SigLIP2) to incorporate multi-label and localization losses [11] [12]. LLaVA bridged the gap to natural language interfaces, showing that vision features plus an LLM can yield open-ended reasoning about images. Modern multimodal models often combine these advances: they use CLIP/SigLIP-style encoders for visual input, and employ LLaVA-style instruction tuning to produce interactive, explainable AI systems. In summary, CLIP provided the core vision-language representation, SigLIP improved its reliability, and LLaVA connected it with language generation – together laying the groundwork for today's VLM architectures and methodologies.

2. Recent Generalist VLMs (State-of-the-Art Multimodal Models)

New large-scale vision-language models have pushed performance across diverse benchmarks. These **generalist VLMs** often combine image, text, and sometimes video understanding in one system. Below is a comparison of several cutting-edge models, highlighting their scale, training, and contributions:

Model	Scale & Base	Training Data	Notable Features	Contributions & Performance
Cambrian-1 (NYU 2024) [13] [14]	13B (LLaMA/ Vicuna backbone) with ~20 vision encoders tested	20+ visual encoders tested; 150K multimodal instructions (public data)	**Vision-centric design:** Explores various vision encoders and connector designs; introduces Spatial Vision Aggregator (SVA) for high-res features; new **CV-Bench** benchmark [15].	Achieves **state-of-the-art multimodal performance**, open-sourcing a "cookbook" for VLM design [16]. Highlights importance of visual encoder choice and data balance.
Qwen2-VL (Alibaba 2024) [17] [18]	2B, 7B, and 72B (Qwen LLM family)	Internal image + video data, multilingual text (Chinese, English, etc.)	**Naive Dynamic Resolution:** Processes arbitrary-resolution images by varying token count [19]. Handles **videos and images in one model**, with Multimodal RoPE for temporal/spatial fusion [20]. Built-in **visual agent** tools and broad multilingual OCR support [21] [22].	**Highly scalable:** 72B model matches GPT-4V and Claude-3.5 on multimodal tasks [23]. Excels in image comprehension (e.g. DocVQA, ScienceQA) and supports 20+ min video Q&A [24] [25]. Sets new benchmarks in dynamic vision-language reasoning.

Model	Scale & Base	Training Data	Notable Features	Contributions & Performance
Molmo (AI2 2024) [26] [27]	7B (transformer, MoE variant) + MoE experts; also 1B efficient model	**PixMo**: 1M curated images with rich human captions + 2D pointing annotations [28]	**Data-efficient strategy:** Uses <1% of data of others via **high-quality captions and pointing** to inject spatial knowledge [28]. Mixture-of-Experts and distilled vision encoder for efficiency. **Can "point" in images** (outputs coordinates to interact with UIs) [29].	**Closes gap to closed models:** 72B Molmo outperforms other open models and approaches GPT-4V, Claude 3.5 on vision tasks [30]. Even a 7B Molmo competes with much larger proprietary models [31]. Demonstrates that *quality trumps quantity* – careful data curation yields strong generalization.
InternVL 2.5 (OpenGVLab 2024) [32] [33]	1B up to 78B (multiple sizes) with InternViT image encoder (up to 6B)	2× data of InternVL 2.0 (filtered for quality); diverse tasks (VQA, grounding, documents, video)	**Progressive scaling:** Train vision encoder with small LLM, then swap in larger LLM – preserves visual skills while scaling up [34]. **Improved training:** Random JPEG augmentation for robustness; loss re-weighting for balanced generation [35]. Focus on **Chain-of-Thought (CoT)** for complex reasoning, with test-time CoT boosting performance [36].	**Open-source rival to GPT-4V:** 78B InternVL2.5 achieves >70% on the challenging MMMU driving benchmark (first open model to do so) [33]. **Matches or surpasses** GPT-4V and Claude 3.5 on many vision-language tasks [33]. Sets new standards in **multilingual understanding and dense prediction**, thanks to data curation and advanced training tricks.

Model	Scale & Base	Training Data	Notable Features	Contributions & Performance
DeepSeek-VL2 (2024) [37] [38]	3B, 16B, 27B *Mixture-of-Experts* LLM (0.6B, 2.4B, 4.1B active params) + Vision enc.	1M+ images (with OCR, charts, VQA annotations); includes text in images	**MoE architecture:** Uses sparse expert LLM (multi-head latent attention) for efficiency – 27B total but ~4B active per token [39]. **Dynamic tiling vision encoder:** Handles **ultra-high-res images** by splitting into tiles, enabling detailed chart/diagram analysis [38]. Strong at **OCR and documents**.	**Superior on structure understanding:** Outperforms others on tasks like document QA, table & chart understanding [40]. More efficient inference due to MoE and latent cache compression, making it faster and lighter despite scale [41]. DeepSeek-VL2 shows that combining LLM reasoning with high-res vision can tackle text-heavy scenes better than conventional VLMs.
Eagle-2 (NVIDIA 2025) [42] [43]	1B, 2B, 9B (Qwen2.5 backbones with SigLIP image encoders)	**180+ data sources** (web images, driving data, text corpora – highly curated) [44] [45]	**Data-centric "post-training":** Rather than inventing new architectures, Eagle2 focuses on **transparent data curation** [42]. Compiles a *Driver's License*-like curriculum of visual tasks, Q&A, and rule understanding. Emphasizes balanced, diverse data (IDK-B, DriveBench, etc.). Fully open process for reproducibility.	**Benchmark leader:** Eagle2-9B matches or beats larger open and even closed models on standard multimodal benchmarks [46]. Offers a **public, reproducible pipeline** for training competitive VLMs with *complete* documentation of data and training. Highlights that careful data engineering can yield reliable and high-performing VLMs without enormous model size.

Model	Scale & Base	Training Data	Notable Features	Contributions & Performance
VideoLLaMA-3 (DAMO 2025) [47] [48]	13B (LLaMA/ Vicuna-based) with video extension	158K high-quality image-text instructions (COCO + GPT-4) + Video instruct data (WebVid, AudioSet) [49] [50]	**Vision-centric training:** Prioritizes **image understanding as the foundation** for video. Trains in stages – (1) adapt vision encoder for variable resolution, (2) massive image-text alignment on scenes, documents, etc., (3) add video fine-tuning for temporal reasoning [50] [51]. Uses **dynamic token merging** to compress similar visual tokens across frames, handling long videos efficiently [48] [52].	**Unified image+video GPT:** Achieves **strong results in both image and video benchmarks** [48]. For example, it's competitive on image QA while also summarizing 20-minute videos [24]. VideoLLaMA-3 proved that a predominantly image-trained model can extend to video with minimal video data, leveraging high-quality static image learning for robust video understanding.

Key Trends: These models show a "**Cambrian explosion**" of VLM design, each extending capabilities in different directions. Several themes emerge:

- **Scaling and Efficiency:** Many use clever strategies to scale up. Qwen2-VL introduces dynamic resolution to handle bigger images without resizing [19]. InternVL2.5 uses progressive training to train 78B models at a fraction of the cost [53]. DeepSeek-VL2's MoE achieves near-27B performance at 4B runtime cost [41]. OpenVLA (from robotics, see below) fuses multiple vision backbones to maximize information [54]. Scaling laws are being explored (Qwen2-VL tested 2B→72B [55]) – larger models tend to perform better, but only when combined with the next point (data).

- **Data Curation and Multimodality:** A shift from pure quantity to **quality and diversity** of training data is evident. Molmo uses *far fewer images* but all with rich human-written descriptions and even spoken annotations [28], yielding strong results with less compute [27]. Eagle-2 curates driving data and rules from 180 sources to explicitly teach traffic knowledge [44] [45]. Many models incorporate **multilingual data** (Qwen2-VL for Chinese/English [17], SigLIP2 for 50+ languages in its recipe). Several now include **video or multi-image** training (VideoLLaMA-3, InternVL2.5, DeepSeek-VL2) so the model can generalize to temporal sequences. The result is VLMs that are not just image captioners, but can handle documents, diagrams, and videos in multiple languages.

- **Integration of Reasoning:** Following the success of LLMs, generalist VLMs often incorporate *reasoning modules or losses*. InternVL2.5 explicitly optimizes and evaluates chain-of-thought responses for complex questions [36]. Qwen2-VL and VideoLLaMA-3 allow multi-step conversations

about visual content. Models like Cambrian-1 and Eagle-2 emphasize interpretability by generating explanations. This trend reflects a push toward **not just perceiving, but also explaining** visual scenes – crucial for high-stakes domains like driving and robotics.

- **Open Source and Adoption:** Importantly, almost all these models are open-sourced (weights or at least code). **OpenVLA** (described in Section 5) sets a strong precedent in robotics by releasing everything, similar to Cambrian-1 and Eagle-2 in the vision-language space [56] [42]. This has accelerated adoption – many research works quickly build on these models (e.g. using Qwen2-VL or OpenVLA as backbones for new tasks). There's also a trend of **community collaborations** (OpenVLA was a joint effort by academia and industry [57]). Overall, the generalist VLMs of 2024-2025 are not only more powerful but also more accessible, fueling rapid progress in both applications and further research.

3. Driving with VLMs: Vision-Language Models for Autonomous Driving

Bringing VLMs into autonomous driving promises more interpretable and generalized decision-making. Several models (often called *DriveVLMs*) have emerged that incorporate vision-language foundations into driving tasks like planning, perception, and explanation. Below we detail key models – their architectures, benchmarks, and how they tackle the challenges of multimodal driving data:

GPT-Driver (2023): This work by Jiang et al. pioneered using an LLM for motion planning. GPT-Driver reformulated **trajectory planning as a language modeling problem** [58]: it represents the driving scene (objects, map, ego state) as text tokens, and has GPT-based models predict the next actions/waypoints as text. By fine-tuning GPT-3.5 on driving trajectories (e.g. from the nuScenes dataset), GPT-Driver could output a sequence of future positions along with a natural language *rationale* for the decision. This gave an element of interpretability ("the car slows down because a pedestrian is crossing"). Studies found that GPT-Driver achieved reasonable planning performance and enhanced decision transparency [59]. However, being open-loop (it produces a plan given an initial scene) and running an LLM in the loop meant it was **computationally heavy and somewhat laggy** for real-time use. Nonetheless, GPT-Driver **pioneered LLM-as-planner** and inspired follow-ups that improve efficiency and closed-loop control.

EMMA (Waymo, 2024): *End-to-End Multimodal Model for Autonomous Driving*. EMMA is a large-scale model that directly maps **raw sensor data to driving decisions** [60]. It uses a multimodal transformer (built on Google's Gemini LLM) to take in **camera images (all around view)** and other inputs (ego speed, route) and output *everything needed for driving*: high-level navigation commands, detected objects, **planner trajectories**, and even road structure predictions [61]. Importantly, EMMA converts these outputs into **natural language tokens** (for example, a trajectory is encoded as a list of coordinates in text form) [62] [63]. By doing so, it leverages the LLM's world knowledge (encoded in text) to enforce traffic rules and common sense. EMMA was trained on Waymo's internal dataset and evaluated on planning benchmarks (nuScenes, Waymo Open Motion) and even 3D object detection. It achieved **state-of-the-art motion planning results** on nuScenes and near-SOTA on detection [64]. Co-training the model on multiple tasks (detection, road layout, planning) yielded a performance boost across all [65] – a testament to the benefit of a **unified multitask model**. EMMA's outputs are more interpretable: one can prompt it for an explanation, since the intermediate representations are language-based. **Limitations:** EMMA currently uses only cameras (no LiDAR) and processes only a few frames at a time, and it requires heavy computation (a 64M parameter

vision backbone + 20B+ LLM) [66]. It's a proof-of-concept that an LLM-based policy can drive a car, at least in simulation, while providing interpretable intermediate outputs (like "stop sign detected, plan to stop"). Waymo's release of EMMA's concepts spurred interest in *LLM-driven end-to-end driving*, as seen in open-source reimplementations (OpenEMMA [67] [68]).

DriveVLM (2024): Proposed by Tsinghua researchers [69], DriveVLM aimed to **merge autonomous driving and large VLMs** by using object-level inputs. It takes the **bird's-eye view (BEV) of the scene encoded as vectors (positions of cars, pedestrians, etc.)** and feeds that into a vision-language model. Essentially, it treats each detected object as a "visual token" and then uses an LLM to reason about what action to take [70]. This approach fuses classic perception (object detection and tracking) with an LLM-based decision module. DriveVLM showed that an LLM can fuse object-level data to perform **explainable control** – for instance, outputting "there is a car ahead, so slow down" along with a low-level control command. It was evaluated in simulation scenarios for decision-making and offered better **interpretability** than black-box neural planners. DriveVLM's convergence of object-centric perception and VLM reasoning is a step toward **modular yet learnable driving** systems. (Notably, a related work *"Driving with LLMs"* by Chen et al. also fused object vectors with LLMs and found improved explainability [70].)

AsyncDriver (ECCV 2024): One major challenge of using LLMs in driving is their slow inference. AsyncDriver (Chen et al. 2024) addresses this with an **asynchronous planner architecture** [71] [72]. It splits the system into two loops: a fast, conventional motion planner (running at say 10Hz) and a slower LLM-based scene interpreter (running at perhaps 1Hz). The LLM (based on GPT-4 or similar) reads the scene (in text form, including route instructions) and produces high-level guidance or "scene-associated instructions" [73]. These might be semantic hints like "there is congestion ahead, prepare to merge left" or safety checks. The real-time planner then takes those into account when computing trajectories. By **decoupling the LLM frequency from the control frequency** [74], AsyncDriver maintains performance comparable to using the LLM every timestep, but with much less computation. In closed-loop simulation on the nuPlan benchmark, AsyncDriver achieved **higher driving scores in complex scenarios** than a planner without LLM guidance [75]. It also preserved safety when the LLM outputs were sporadic or delayed. This model shows a practical way to **inject reasoning into a live driving system** without sacrificing responsiveness: use the LLM's advice asynchronously. It still faces the reliability of the LLM (it must not give wrong advice), but by keeping the human-tuned planner in charge of actual control, it adds interpretability and common-sense without full reliance. AsyncDriver's concept of adjustable LLM inference frequency could be extended to any real-time system that wants occasional "brainy" guidance without constant overhead [76] [77].

DriveMM (Dec 2024) – *All-in-One Multimodal Model* (Huang et al.). This is an academic take on a **generalist driving model** similar in spirit to EMMA. DriveMM trained a single transformer on **diverse driving datasets (CARLA, nuScenes, Waymo, etc.) across multiple tasks: perception, prediction, planning** [78] [79]. It introduced a *curriculum learning* schedule: first train on simpler visual comprehension tasks (e.g. object recognition) then progressively include harder tasks like planning [79] [80]. The model ingests multi-view images (like 6 camera views around the car) and outputs a unified token sequence that can be decoded into detections, trajectory plans, etc. Thanks to multi-task training, DriveMM achieved **state-of-the-art on 6 public benchmarks** (including detection and motion prediction) and showed strong zero-shot transfer to a new dataset [81] [82]. For example, without fine-tuning it could drive in a simulator it never saw during training. This indicates the model learned some general driving *common sense*. DriveMM's significance is in demonstrating that a **single multimodal model can match specialized models** in each subtask, suggesting that in the future, autonomous vehicles might run one large neural model for the whole pipeline. It also underscored the value of *cross-domain training*: by training on varied cities and conditions,

the model generalized better. The authors open-sourced their code, contributing to the growing trend of **open end-to-end driving research**.

DIMA (Jan 2025) – *Distilling Multimodal LLMs for Driving* (Hegde et al.) is a noteworthy attempt to **compress an LLM-enhanced planner into a smaller, efficient model** [83] [84]. DIMA starts with a multimodal LLM (similar to GPT-Driver or EMMA's backbone) that can plan safe trajectories but is too slow for real use. They then perform **knowledge distillation**: create a vision-only planner (student) and train it to mimic the decisions of the big VLM planner (teacher) on a variety of driving scenarios [85]. To do this, they set up *surrogate tasks* that force the student to align with the teacher's internal reasoning. For instance, the student is trained to predict not just the next action but also intermediate "explanation tokens" the teacher would produce [86] [87]. The result, DIMA, is an end-to-end vision-based driving model that **doesn't require an LLM at inference** but has absorbed much of the LLM's traffic knowledge and foresight [88]. Impressively, DIMA reduced trajectory error by 37% and collisions by 80% compared to the original vision-only model [84], nearly closing the gap to the LLM planner. It also achieved **state-of-the-art on nuScenes planning benchmark** [84]. DIMA's approach combines the best of both worlds: the **world knowledge and interpretability of LLMs** with the **speed of a compact network**. This hints that even if LLMs themselves aren't deployed in the car, they can be used offline to teach smaller driving models how to handle rare or complex events (the "long-tail" scenarios). DIMA and similar distillation efforts increase trust in autonomous driving models by maintaining efficiency *and* robustness.

SENNA (ICCV 2024): *Senna* by Jiang et al. takes a **hybrid approach** to join an LVLM with a conventional end-to-end driving model [89] [90]. They observed that LVLMs are great at reasoning but not precise in output (you wouldn't trust GPT-4 to output exact steering angles), whereas end-to-end driving nets are precise but lack commonsense. SENNA therefore consists of **Senna-VLM + Senna-E2E**: the VLM looks at multi-camera images and *outputs a high-level plan in text* ("Slow down and prepare to turn left at the intersection") [91] [89]. The E2E module (a CNN controller) then takes this plan plus the images and produces the exact trajectory. Essentially, SENNA *"decouples high-level planning from low-level control"* [89]. During training, they use planning-oriented Q&A to tune the VLM to traffic decisions (e.g. asking it what the correct action is in a scene) [92], and a three-stage training curriculum so the VLM learns general road reasoning first, then refines on specific driving data [93]. SENNA achieved **state-of-the-art planning performance** on two driving datasets, and with additional large-scale pretraining (DriveX dataset) it cut planning error by 27% and collisions by 33% vs. no-pretrain [94] [95]. The key insight is that SENNA's language outputs act as an **interface**: you can inspect or even edit the intermediate plan. By not forcing the LVLM to output raw control, it avoids the weakness of LVLMs in numeric precision [91] [96]. SENNA points toward a future where an AI driver might literally "think out loud" in natural language ("I will yield to the pedestrian") and then execute the action. This makes debugging and validation easier, and leverages the strength of each component (language for reasoning, network for control). A similar philosophy is echoed by AsyncDriver and DIMA – combine interpretable reasoning with reliable low-level execution.

Summary of Driving VLMs: The above models illustrate how vision-language techniques are being applied to autonomous driving. Two broad approaches exist: (1) **LLM-based planners (GPT-Driver, DriveVLM, AsyncDriver, SENNA)**, which use language as an intermediate for planning/explanation, and (2) **Multimodal end-to-end networks (EMMA, DriveMM, DIMA)**, which incorporate language or LLM knowledge into a direct sensor-to-action model. Across the board, benchmarks used include **nuScenes** (for trajectory planning and perception), Waymo Open Dataset (motion prediction), and bespoke simulation tests (e.g. CARLA driving tests). Many works report metrics like collision rate, off-road rate, or planning error to quantify performance and safety. The trend is that introducing VLM/LLM components often **improves**

interpretability and handling of rare events, but one must mitigate issues of speed and reliability. Techniques like asynchronous processing (AsyncDriver) and distillation (DIMA) are actively addressing these concerns. Driving-focused VLMs also highlight multimodal data handling: they take images, maps, and even LiDAR (in some cases) along with textual instructions (destination, traffic rules) as input – truly **multimodal sensor fusion** via language models. While these systems are largely in research or simulation phases, they demonstrate the potential of VLMs to make autonomous driving more **transparent, rule-aware, and adaptable**. In coming years, we can expect further integration of such models into real AV stacks, especially as their real-world performance and validation (safety assurances, out-of-distribution handling) improve.

4. Driving with VLM Benchmarks (Evaluation and Readiness)

To understand whether VLMs are viable for autonomous driving, researchers have begun establishing **benchmarks and evaluation frameworks** focusing on reliability, safety, and "driving intelligence" of these models. Unlike generic vision benchmarks, driving-oriented tests examine how well models follow traffic rules, handle corruptions, and explain decisions. Here we discuss key benchmarks and studies:

DriveLM / "Driving with LLMs" Benchmarks (2023): Early explorations like *Driving with LLMs* assembled evaluation sets to see if LLM-based planners make sensible decisions. One such benchmark fed object-level scene info to an LLM and asked it questions like *"What should the ego car do next?"* or *"Is it safe to turn right now?"*. While not a formal name, these efforts highlighted the need for systematic testing. They found LLM planners can answer straightforward questions but struggled with complex multi-step reasoning in dense traffic [97] . These pilot studies led to more structured benchmarks described below.

"Are VLMs Ready for Autonomous Driving?" (DriveBench, 2025) [98] [99] – Xie et al. introduce **DriveBench**, an extensive empirical study of VLMs in driving contexts. They specifically evaluate *VQA-style performance* and visual grounding reliability under various conditions. DriveBench includes **19,200 frames** from driving scenes and **20,498 QA pairs** about those scenes, spanning 4 driving tasks and 3 question types [100] . Crucially, it tests models in **17 different input conditions**: clean images, images with corruptions (fog, motion blur, etc.), and even text-only (no image) where the model sees just the question [101] [102] . Twelve popular VLMs (both general models like BLIP-2 and driving-specific ones) are evaluated. The findings are sobering: **current VLMs often give plausible-sounding answers that are not truly grounded in the visual input** [103] . For example, a model might "hallucinate" a response based on context (e.g. assuming a traffic light is green because of prior knowledge, even if the image is unclear) [97] . Under degraded visuals, this got worse – models would still output confident answers even with missing or corrupted images [104] . This exposes a **reliability risk**: VLMs might appear to understand a driving scene but actually rely on priors or biases. The authors also noted multi-modal reasoning (combining map info + image, etc.) was weak [105] , and model performance was **inconsistent and brittle to noise**. To address this, they propose improved evaluation metrics that explicitly reward correct visual grounding. They suggest checking if the model's answer changes appropriately when the image is altered – if not, the model might be answering from expectation rather than perception. They even leverage the model's own awareness: interestingly, VLMs can often detect that an image is corrupted (outputting "the image is blurry"), so that could be used to decide when to trust the model's content answers [106] . Overall, this benchmark concludes that **VLMs are not yet "driver's license ready"** – they need more rigorous grounding and consistency for safety-critical use [107] . DriveBench and similar efforts provide researchers a tool to quantify progress on these fronts as new models (like those in Section 3) are developed.

"Can VLMs Obtain a Driver's License?" (IDKB, 2024) [108] [109] – Lu et al. created the **IDKB** (Interactive Driving Knowledge Base) dataset as a step-by-step "driver's test" for LVLMs. They observed that most vision-language driving datasets focus on perception ("what is in the scene?") or simple decision Q&A, but none explicitly teach or test **traffic rules and driving theory** [110] [111]. IDKB fills this gap by compiling over **1 million data items** covering: official driving handbooks, exam questions from multiple countries, and even simulated road scenarios with rule-based queries [109]. It's essentially the textual knowledge a human would need to pass a driving written test, plus applied scenario Q&A (like "At a four-way stop, who has right of way?"). The authors then took 15 large VLMs and **grilled them on IDKB**. The results showed that out-of-the-box LVLMs (even powerful ones) **lack specialized driving knowledge** [110] [111] – they often fail detailed rule questions or get tricked by nuanced scenarios, because their general training didn't include those specifics. However, after fine-tuning some models on IDKB, their performance greatly improved [112]. This suggests that to "earn a driver's license," an AI must be explicitly taught driving rules, not just expected to learn them from generic web data. The IDKB benchmark is a step toward **reliable AGI for driving** – combining perceptual ability with the rule-based knowledge that human drivers internalize. It also emphasizes evaluating **safety compliance**: a model might recognize cars and pedestrians well, but if it doesn't know it must stop for a school bus, that's a critical failure. By publishing IDKB and initial model scores, this work provides a yardstick for measuring how close a VLM is to having the knowledge of a licensed driver (in theory). It encourages the community to incorporate explicit driving knowledge into model training (as Eagle-2 and others have begun to do) [44] [109].

TOD3Cap (2024) – 3D Dense Captioning for Driving Scenes: This benchmark (Jin et al.) is a bit different – it focuses on **comprehensive scene understanding**. TOD3Cap stands for *"Towards Outdoor 3D Dense Captioning"* [113]. The task is: given a driving scene with multi-modal sensor input (LiDAR point cloud + surround RGB images), detect all the important objects and output a descriptive caption for each [114] [115]. For example, the model should produce: *"Car on the left, red sedan waiting at intersection"; "Pedestrian crossing the street"; "Traffic light showing green"*, each tied to a 3D location (bounding box). This is extremely challenging because it requires both detection and fine-grained description in an open-ended way. TOD3Cap introduced a dataset of **850 driving scenes** with **64.3k objects and 2.3 million captions** describing them [116] – the largest of its kind for outdoor scenes. They also proposed a model that combines BEV detection with a Q-Former + LLaMA-Adapter to generate captions for each object [115] [117]. This model significantly outperformed baseline methods (improving dense captioning score by +9.6 CiDER) [118] [119]. The benchmark tests how well models can **not only identify objects, but contextualize them** (for instance, note attributes or actions). Applications include generating rich scene descriptions for HD map annotation, VQA, or even feeding detailed information to planning modules. TOD3Cap touches on the **explainability** aspect of driving AI – a system that can densely caption might be used to justify its perception ("I see a cyclist approaching from behind") in natural language. It's also a step toward unifying 2D and 3D understanding: models must fuse camera and LiDAR to caption accurately. As a benchmark, TOD3Cap will push VLMs to have stronger multi-object and multi-modal grounding. Success on TOD3Cap means a model has a *holistic understanding* of a driving scene, which correlates with driving competency. Already, some works (e.g. DexVLA in robotics) cite TOD3Cap to emphasize the need for detailed scene understanding in planning [120].

AutoTrust (2024) – Trustworthiness Benchmark [121] [122]: Perhaps the most comprehensive **safety and ethics evaluation** for DriveVLMs, AutoTrust by Xing et al. is a benchmark specifically targeting the **trustworthiness** of driving VLMs [121]. They define five dimensions: **Trustfulness** (no hallucinations or lies), **Safety** (no advice that causes accidents, obeying rules), **Robustness** (resilience to adversarial inputs or perturbations), **Privacy** (not revealing sensitive info like license plates or faces), and **Fairness** (no biased

behavior or disparate performance) [121] [123]. AutoTrust constructed a visual QA dataset of driving scenes with **10k unique scenes and 18k queries** designed to probe these aspects [123] [124]. For example, some queries deliberately ask for sensitive information ("What is the license number of the car ahead?") to test privacy, or present manipulated images (like a stop sign with a sticker) to test robustness. They evaluated six advanced VLMs (both generalist like GPT-4V and specialist like DriveLM-Agent). The results revealed **previously undiscovered vulnerabilities** [125]. Notably, some *general-purpose* models (LLaVA 1.5, a mini GPT-4 Vision) were actually **more trustworthy overall** than models fine-tuned for driving [126]. For instance, a driving-specialized model might overfit to training and blurt out a license plate (privacy breach), whereas a general model refuses. One model, DriveLM-Agent, was found to frequently **disclose sensitive info** (it wasn't trained to censor it) [127]. Both general and driving models were **susceptible to adversarial attacks** – e.g. a small sticker on a stop sign might make them fail to mention the stop sign, or a question phrased a certain way could trick them into dangerous advice. Bias/fairness issues were also present: models performed worse or differently in scenes from different regions or with different pedestrian demographics, indicating potential bias [128]. These findings ring alarm bells that even if a DriveVLM performs well on standard metrics, it might harbor unsafe behaviors. AutoTrust provides a **benchmark to quantify these risks** and track improvements as new models address them. The authors released the dataset and a leaderboard [129], calling for the community to **take immediate action to improve DriveVLM trustworthiness** [130]. This has spurred research into methods like safer training (filtering sensitive content), robustness training (adversarial image augmentations), and better grounding (to reduce hallucination). In short, AutoTrust is about answering: *Can we trust this driving AI?* – not just to be correct, but to be safe, secure, and fair. It's a crucial complement to performance benchmarks, ensuring that progress in capability comes with progress in reliability.

Trends in Benchmarking: The emergence of these benchmarks shows a maturation in the field – moving from demonstrating capability to **ensuring reliability and accountability**. We see benchmarks tackling: **interpretability** (DriveBench QAs if explanations truly reflect images), **explicit knowledge** (IDKB's driver's ed content), **holistic perception** (TOD3Cap's dense captions), and **ethical/safety principles** (AutoTrust). An *"open research question"* highlighted by these works is how to quantify and improve **visual grounding**. Both DriveBench and AutoTrust found models often rely on textual cues or hallucinate; improving grounding might involve new training losses or architecture (e.g. forcing alignment between what the model says and image evidence). Another open question is **multi-modality of input** – many benchmarks focus on images, but driving involves LiDAR, maps, odometry. Some works (like TOD3Cap) are starting to incorporate that. We can expect future benchmarks to include more sensor types and perhaps even closed-loop driving tests (evaluate a model by putting it in a simulator and seeing if it crashes). Also, **standardizing evaluation metrics** is in progress: for instance, AutoTrust proposes metrics like *Demographic Accuracy Difference* for fairness and worst-case accuracy for robustness [131] [132]. These efforts will help researchers objectively compare models and ensure that a "better" model truly means better not just at answering questions, but at doing so correctly and safely.

In summary, benchmarking VLMs for driving is quickly extending beyond traditional accuracy metrics to include **safety, ethics, and domain-specific knowledge**. This is a positive development, as it will guide the community to address the critical question: not just *"Can a VLM drive?"* but *"Can it drive well and be trusted like a human driver?"*. The current answer is "not yet" – but thanks to these benchmarks, we have a clearer map of what needs to improve to get there.

5. VLMs in Robotics and Beyond (Cross-Domain Applications)

Vision-Language Models are increasingly being applied to robotics, where an agent must see, reason, and act. In robotics, VLMs (sometimes called **Vision-Language-Action models, VLAs**) are used for tasks like manipulation, navigation, and human-robot interaction. Here we explore three notable projects – **Pi·0**, **OpenVLA**, and **DexVLA** – and how they extend VLM concepts to robotics, comparing their approaches to non-robotic VLM use cases:

Pi·0 (π0) – General Robot Control via VLM (Physical Intelligence, 2024): Pi·0 is a **vision-language-action flow model** for general robotic control [133]. It represents one of the first attempts to train a *foundation model* for robotics akin to GPT-3 for text. The approach uses a transformer backbone (initialized from a large VLM like PaLI or similar) and adds an *action prediction head*. Unlike static VLMs, Pi·0 is trained on **robot trajectories**: sequences of images, instructions, and corresponding robot actions (e.g. joint commands) across many tasks and robot types [134] [135]. Importantly, Pi·0 uses a technique called **flow matching** (or trajectory matching) to learn the mapping from visual state to actions. Rather than training via trial-and-error reinforcement learning (which is slow), it treats the problem as a supervised sequence prediction – much like language modeling, but for motor commands. A challenge Pi·0 tackled was **action space representation**: continuous robot motions need to be discretized for a transformer. The team developed a compression method (called FAST tokenization) using discrete cosine transforms to efficiently encode high-frequency control signals as compact tokens [136] [137]. This allowed Pi·0 to ingest, for example, 1000 Hz control signals without an explosion of sequence length. With FAST tokens, they trained **π0-FAST**, a variant of Pi·0, on multi-robot data (including mobile robots, manipulators, and even dexterous hand tasks) [138] [139]. The result was a **generalist policy** that could, zero-shot or with minimal fine-tuning, control different robots to perform tasks, often rivaling specialist policies. Pi·0's performance on a suite of tasks approached that of the best diffusion model policies (which were another state-of-the-art) but with much faster training and inference [139]. In essence, Pi·0 demonstrated that a single VLA model can learn **cross-embodiment skills** – e.g. knowledge like "grasping an object" or "avoiding obstacles" that transfer between a wheeled robot and a robotic arm – by leveraging the sequence modeling power of transformers. Compared to non-robotic VLMs, Pi·0 had to deal with *temporal coherence* and *physical constraints* in outputs (making sure generated actions are smooth and valid). Its success showed that techniques from NLP (like tokenization and transformer scaling) can be adapted to robotics, leading to more **"LLM-like" robot brains**. Pi·0's open policy (the company hinted at releasing model or benchmarks) has influenced academic work on open robot foundations (e.g. OpenVLA).

OpenVLA (2024) – *Open Vision-Language-Action Model* [56]. A collaboration between Stanford, Berkeley, TRI, and Google, OpenVLA is essentially the **CLIP+GPT of robotics** – a 7B-parameter multimodal model that takes vision and language input and outputs high-level robot actions [56] [140]. It was trained on an unprecedented dataset: **970k real robot episodes** from the Open X-Embodiment dataset [56], which includes demonstrations from many robots (robotic arms, mobile manipulators, etc.) performing diverse tasks. OpenVLA's architecture has three main parts [54]: (1) a **fused visual encoder** that combines **SigLIP** (for image-language alignment) and **DINOv2** (for strong visual features) to encode camera inputs [141]; (2) a projection module to map visual embeddings into the LLM's token space; and (3) a **pretrained language model (Prismatic-7B)** that has been adapted to output a sequence of actions given the text+image inputs [141]. During training, the "language" the LLM outputs is actually a sequence of discrete actions (like a token representing "gripper close" or "move end-effector up") that correspond to the demonstration data. By open-sourcing this model, the authors provided the community its first **open generalist robot policy model**. *Contributions:* OpenVLA establishes a new **state of the art in general robot manipulation** – it can

control multiple robot types out-of-the-box and can quickly adapt to new ones via fine-tuning [142] [143]. For instance, with minimal additional data, OpenVLA was fine-tuned to a new robot arm and achieved complex tasks like stacking blocks and tool use, matching performance of specialist models. It also supports **multi-modal prompts**: one can give it a textual instruction (e.g. "pick up the red block and put it on the green block") along with camera images, and it will generate the action sequence to do it [144] [145]. This is analogous to how a VLM like Flamingo can take an image and a question and answer in text – except OpenVLA answers in *robot actions*. Compared to non-robotic VLMs, OpenVLA had to incorporate **embodiment knowledge**: it added proprioceptive state inputs and had multi-headed action outputs to handle different robot kinematics [146]. Essentially, it learned a *unified representation of both vision and motor commands*. In non-robotic settings, VLMs typically output text or labels, but here the output is something that *directly causes change in the physical world*. OpenVLA's success (and the fact that DeepMind/Google participated) underscores how important bridging vision, language, and action has become – it's the path to more capable home robots and industrial automation. It also shows cross-domain influence: OpenVLA leverages advances from vision (SigLIP features [54]), language (LLaMA-based LLM), and even techniques like LoRA fine-tuning for quick adaptation, bringing them all into the robotics domain.

DexVLA (2025) – *Dexterous Vision-Language-Action with Diffusion* (Wu et al. 2025) [147] [148]. DexVLA is a cutting-edge framework aimed at **complex, long-horizon tasks and dexterous manipulation**. It introduces a hybrid model that combines a **VLM-based reasoning module with a diffusion policy expert** [149] [150]. The architecture (summarized earlier) works as follows: an image encoder embeds multi-camera views into tokens; a transformer (LLM) processes those along with the text command, and outputs two sets of tokens – "reasoning tokens" and "action tokens" [151] [152]. The *reasoning tokens* are like the model's internal thought process (in latent form), and they are injected (via FiLM layers) into a *Diffusion Policy* model which actually generates the sequence of low-level actions [152] [153]. The *action tokens* from the LLM are also passed to this diffusion model as a guide. Essentially, DexVLA splits the problem: the LLM part handles high-level planning and understanding (and can be trained with language supervision), while the diffusion part handles **fine-grained motor control** with continuous actions (since diffusion models excel at modeling complex continuous distributions) [154] [155]. They also introduced a 3-stage *embodied curriculum learning* for training: Stage1, pre-train the diffusion policy on many robots (cross-embodiment) to learn general motor skills; Stage2, fine-tune diffusion on the specific robot; Stage3, fine-tune the combined system on specific tasks with the reasoning module active [156] [157]. By doing so, DexVLA achieved the ability to perform **versatile, long-horizon tasks** (like a robot hand manipulating objects through multiple sub-goals) that pure VLM or pure diffusion approaches struggled with [148]. In evaluations, DexVLA proved capable of handling **complex dexterous manipulation** and multi-step procedures by leveraging the strengths of both components [148]. For example, in a kitchen cleanup scenario, the LLM can reason "first pick up the plate, then move to the sink, then scrub it" while the diffusion expert precisely controls the arm to execute each step. Compared to non-robotic VLMs which might output a paragraph of text for an answer, DexVLA's "answer" is a sequence of actions – but notably, it also produces an internal reasoning which could be decoded or visualized for interpretability (the FiLM-injected reasoning is analogous to chain-of-thought). DexVLA's *plug-in diffusion expert* idea is somewhat analogous to how some multimodal models plug in an OCR module or a calculator into an LLM – here the diffusion model is like a specialist tool for action generation. This modular design might influence future robotics VLMs to incorporate various "experts" (for grasping, navigation, etc.) that an LLM can coordinate. In terms of impact, DexVLA represents the **state-of-the-art integration of decision-making and control**: it provides a promising approach for robots to handle **open-ended instructions (thanks to VLM reasoning) with precise execution (thanks to diffusion)** [148]. It also shows cross-pollination: diffusion models were first popular in image generation;

here that concept is repurposed to generate robot trajectories, guided by language – truly a marriage of diverse AI techniques.

Robotics vs Non-Robotic VLM Use Cases: The above models (Pi·0, OpenVLA, DexVLA) highlight both similarities and differences with "static" VLMs:

- **Embodied Output:** Unlike captioning or VQA models, robotics VLMs output *actions*. This means their evaluation is on success rates of tasks, not just accuracy of an answer. There's an added feedback loop – wrong outputs lead to physical failure, not just a wrong label. Thus, robotic VLMs often incorporate *closed-loop training or simulation testing*. For instance, Pi·0 and OpenVLA were tested on actual robots or simulators performing tasks, and DexVLA uses diffusion which naturally integrates over a sequence.

- **Cross-Domain Generalization:** In robotics, generalization means across different hardware (embodiments) and different tasks. OpenVLA's success in multi-robot control is akin to a single vision model working for images, videos, and documents – a kind of generality that's even harder because each robot has different dynamics. Techniques like multi-head outputs for each robot (in DexVLA) and progressive training from general to specific (InternVL2.5 in vision, DexVLA in robots) are common to achieve generality [155] [156]. Non-robotic VLMs also aim for generality (one model for many vision tasks), but robots add another layer (generalizing over embodiments and environments).

- **Role of Simulation:** Robotics often uses simulators to generate large datasets (e.g. IDKB or DriveX for driving, or multi-object simulation for DexVLA's Stage1). This is analogous to using synthetic data for vision (like GPT-4 generated image captions as in LLaVA [7]). Both fields leverage synthetic data, but for robots it's crucial to then adapt to real-world data (sim2real gap). OpenVLA's use of 970k *real* episodes is a milestone – akin to collecting a giant real image dataset for ImageNet, indicating the field is moving toward real-data at scale.

- **Interactivity and Adaptation:** In non-robotic scenarios, a VLM's job might end at giving an answer. In robotics, the model's output changes the world state, so it often needs to **observe the result and continue the task**. That means robotics VLMs are designed to work in **interactive loops**. For example, Pi·0 can be run step by step in a control loop, DexVLA's diffusion outputs a whole trajectory but the model could re-plan if something unexpected happens. This interactive aspect is being explored in vision also (think of an interactive VQA agent that can ask follow-up questions or request another view), but it's inherent in robotics. As a result, robotics models sometimes incorporate **feedback mechanisms** or error recovery strategies that static VLMs don't have to consider.

Impact on Robotic Decision-Making: The introduction of VLMs to robotics has made robots more **adaptive and semantic**. Rather than hard-coding every motion, robots with VLM brains can understand high-level goals ("clean up the kitchen") and figure out sub-tasks, potentially even handle new objects by description. This is a huge step towards **general-purpose robots**. Models like OpenVLA and DexVLA show that if you train on enough varied data, the robot can **recombine known skills to perform novel tasks**, similar to how GPT-3 can write a new story by combining patterns seen in training. They also make robots more **language-controllable** – a user can instruct a robot in natural language, and the VLM will translate that to actions (OpenVLA demonstrated this via prompts [158]). This lowers the barrier to human-robot interaction.

Adaptability to Different Environments: Robotics VLMs are being tested in many environments – kitchens, factories, driving in cities, etc. The idea of *cross-embodiment* (like Pi·0 and DexVLA's Stage1) is to learn abstract skills that apply anywhere (e.g. balancing an object, or yielding to pedestrians). Early results are promising: e.g. DexVLA's approach reduced data needs by 60% for a new robot by leveraging pretraining on others [159] [157]. This mirrors how a vision model pretrained on ImageNet can quickly adapt to medical images. In robotics, *similarly, a VLM pretrained on diverse devices can adapt to a specific device much faster* – an efficiency gain crucial for deploying robots in new settings.

Comparison to Non-Robotic Use Cases: Outside robotics, VLMs mostly serve to interpret or generate descriptions (e.g. captioning, retrieval, reasoning about images). In robotics, the **stakes are higher** – the decisions directly affect physical safety and task success. Interestingly, we see a convergence: techniques from robotics VLMs (like **grounding actions with diffusion or optimization**) could benefit non-robotic VLMs. For instance, a VLM that plans a sequence of image edits or web navigation steps is quite analogous to a robot planning physical steps; the diffusion policy idea could be applied to such multi-step AI planning tasks. Conversely, work in vision grounding and language reasoning (as assessed by DriveBench or AutoTrust) feeds back into robotics – a robot must have rock-solid grounding to avoid mistakes. Thus, research in these domains is mutually enriching.

In conclusion, **VLMs in robotics** like Pi·0, OpenVLA, and DexVLA are pushing AI beyond understanding images to taking actions in the real world. They leverage cross-domain knowledge (vision + language + control) to create more general and capable robots. These models are **impacting robotic decision-making** by providing common-sense reasoning, easier programmability via language, and the ability to learn from vast prior data rather than task-specific programming. They highlight an exciting future where the lines blur between "seeing" and "doing" – an AI that can both perceive its environment and interact with it, guided by the rich semantic understanding encoded in vision-language representations. This is the natural next step beyond static VLMs: *embodied AI* that learns and operates in our multi-modal world, bridging the gap from recognition to action. As these robotic VLMs improve, we'll see more adaptable, intelligent machines in homes, hospitals, and workplaces, using the same core models that also understand images and language in non-embodied contexts. The synergy between robotics and non-robotic VLM advancements will likely accelerate progress in both arenas, ultimately leading to AI systems that are truly general-purpose across domains.

Sources: [2] [5] [4] [3] [7] [9] [17] [20] [30] [28] [33] [36] [40] [46] [45] [64] [66] [72] [75] [84] [89] [93] [97] [109] [116] [125] [56] [54] [152] [148]

---

[1] [2] [10] [11] [12] arxiv.org
https://arxiv.org/pdf/2502.14786

[3] [4] Google's SigLIP: A Significant Momentum in CLIP's Framework
https://www.analyticsvidhya.com/blog/2024/10/googles-siglip/

[5] SigLIP Classification Model: What is, How to Use
https://roboflow.com/model/siglip

[6] [7] [8] [9] LLaVA
https://llava-vl.github.io/

13 14 15 16 [2406.16860] Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs
https://arxiv.org/abs/2406.16860

17 Qwen2-VL
https://huggingface.co/docs/transformers/en/model_doc/qwen2_vl

18 19 20 23 55 [2409.12191] Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution
https://arxiv.org/abs/2409.12191

21 22 24 25 Qwen2-VL
https://qwen2.org/vl/

26 27 28 29 30 31 Introducing Molmo: A Family of State-of-the-Art Open Multimodal Models | Business Wire
https://www.businesswire.com/news/home/20240925326133/en/Introducing-Molmo-A-Family-of-State-of-the-Art-Open-Multimodal-Models

32 33 34 35 36 53 InternVL2.5
https://internvl.github.io/blog/2024-12-05-InternVL-2.5/

37 38 39 40 41 DeepSeek-VL2: Mixture-of-Experts Vision-Language Models for Advanced Multimodal Understanding
https://arxiv.org/html/2412.10302v1

42 43 GitHub - NVlabs/EAGLE: Eagle Family: Exploring Model Designs, Data Recipes and Training Strategies for Frontier-Class Multimodal LLMs
https://github.com/NVlabs/EAGLE

44 45 46 Eagle 2: Building Post-Training Data Strategies from Scratch for Frontier Vision-Language Models
https://arxiv.org/html/2501.14818v1

47 48 49 50 51 52 [2501.13106] VideoLLaMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding
https://arxiv.org/abs/2501.13106

54 56 57 140 141 142 143 OpenVLA: An Open-Source Vision-Language-Action Model
https://openvla.github.io/

58 69 70 GitHub - ge25nab/Awesome-VLM-AD-ITS: This repository collects research papers of large Vision Language Models in Autonomous driving and Intelligent Transportation System. The repository will be continuously updated to track the latest update.
https://github.com/ge25nab/Awesome-VLM-AD-ITS

59 Towards Safe Autonomous Driving via Personalized Safety-Critical ...
https://arxiv.org/html/2502.15119v1

60 61 62 63 64 65 66 EMMA: End-to-End Multimodal Model for Autonomous Driving
https://waymo.com/research/emma/

67 Open-Source Multimodal Model for End-to-End Autonomous Driving
https://arxiv.org/abs/2412.15208

68 taco-group/OpenEMMA - GitHub
https://github.com/taco-group/OpenEMMA

71 72 73 74 75 [2406.14556] Asynchronous Large Language Model Enhanced Planner for Autonomous Driving
https://arxiv.org/abs/2406.14556

76 77 awesome LLM for Autonomous Driving resources - GitHub
https://github.com/Thinklab-SJTU/Awesome-LLM4AD

78 79 80 81 82 [2412.07689] DriveMM: All-in-One Large Multimodal Model for Autonomous Driving
https://arxiv.org/abs/2412.07689

83 84 85 86 87 88 [2501.09757] Distilling Multi-modal Large Language Models for Autonomous Driving
https://arxiv.org/abs/2501.09757

89 90 91 92 93 94 95 96 [2410.22313] Senna: Bridging Large Vision-Language Models and End-to-End Autonomous Driving
https://arxiv.org/abs/2410.22313

97 98 99 100 101 102 103 104 105 106 107 [2501.04003] Are VLMs Ready for Autonomous Driving? An Empirical Study from the Reliability, Data, and Metric Perspectives
https://arxiv.org/abs/2501.04003

108 109 110 111 112 [2409.02914] Can LVLMs Obtain a Driver's License? A Benchmark Towards Reliable AGI for Autonomous Driving
https://arxiv.org/abs/2409.02914

113 114 115 116 117 118 119 [2403.19589] TOD3Cap: Towards 3D Dense Captioning in Outdoor Scenes
https://arxiv.org/abs/2403.19589

120 "TOD"^{3}Cap: Towards 3D Dense Captioning in Outdoor Scenes - arXiv
https://arxiv.org/html/2403.19589v1

121 122 123 124 125 126 127 128 129 130 132 AutoTrust : Benchmarking Trustworthiness in Large Vision Language Models for Autonomous Driving
https://arxiv.org/html/2412.15206v1

131 Explainable AI for Safe and Trustworthy Autonomous Driving
https://www.researchgate.net/publication/384911754_Explainable_AI_for_Safe_and_Trustworthy_Autonomous_Driving_A_Systematic_Review

133 134 (PDF) Diffusion-VLA: Scaling Robot Foundation Models via Unified ...
https://www.researchgate.net/publication/386454973_Diffusion-VLA_Scaling_Robot_Foundation_Models_via_Unified_Diffusion_and_Autoregression

135 Deeper Dive into π0Live my life consciously - Lumen's Notes.
https://www.lumeny.io/papers/Deeper-Dive-into-Pi-0

136 137 138 139 physicalintelligence.company
https://www.physicalintelligence.company/download/fast.pdf

144 OpenVLA - NVIDIA Jetson AI Lab
https://www.jetson-ai-lab.com/openvla.html

145 158 OpenVLA – AI Agent Index
https://aiagentindex.mit.edu/openvla/

146 147 148 150 151 152 153 154 155 156 157 159 DexVLA: Vision-Language Model with Plug-In Diffusion Expert for General Robot Control
https://arxiv.org/html/2502.05855v1

149 (PDF) DexVLA: Vision-Language Model with Plug-In Diffusion ...
https://www.researchgate.net/publication/388883804_DexVLA_Vision-Language_Model_with_Plug-In_Diffusion_Expert_for_General_Robot_Control