# A Multimodal AI System for Animal Transport Mode Planning

Yazan Alnakri

November 24, 2025

**Abstract**

This report presents a multimodal artificial intelligence system designed to determine appropriate transportation modes for animals based on their species, size, domestication status, danger level, and geographic travel distance. The system integrates a vision-language model (VLM) for animal classification and a fine-tuned language model (LLM) with task-aware training for structured transport reasoning. **The two-module design separates perception and reasoning primarily to make the training pipeline transparent—only the LLM requires fine-tuning—although a sufficiently capable VLM could, in principle, perform the entire task end-to-end.** Evaluation covers task accuracy, structural validity, and reasoning quality. The project demonstrates the ability to make a complete, modular AI engineering pipeline combining synthetic dataset generation, multimodal inference, LoRA finetuning, and rigorous evaluation, inspired by recent advances in multimodal large language models [1, 2, 3, 4, 5, 6] and structured-output instruction tuning [7, 8, 12, 13]. Code available at: `https://github.com/Yazangthb/animal_transport_project`.

## 1 Introduction

Animal transport involves strict safety, welfare, and regulatory constraints that vary based on species, size, and potential danger to humans. Traditional methods require expert reasoning and specialized knowledge. This project proposes a multimodal AI system capable of automatically recommending safe transportation modes using a combination of visual understanding and structured reasoning.

Recent work on multimodal large language models (MLLMs) has demonstrated that combining powerful visual encoders with language models enables rich image-grounded reasoning and dialogue [1]. Models such as CLIP [2], Flamingo [5], BLIP-2 [6], LLaVA [3], and Qwen-VL [4] show that a generic vision-language stack can be adapted to diverse downstream tasks with limited task-specific data. In parallel, instruction-tuned LLMs have become strong zero-shot learners for structured outputs like JSON and XML [8, 7, 13].

In transport and logistics, AI systems increasingly act as decision support tools for mode selection and routing. For example, masked language models have been used to model transport mode choice behavior [14], reinforcement learning has been applied to dynamic mode selection under congestion [15], and hybrid recommender + multi-criteria decision systems have been used to recommend transport modes under explicit constraints [16]. Our problem of animal transport mode planning is closely related but introduces additional safety and welfare constraints tied to animal categories and danger levels.

The system follows a three-stage architecture:

1. **Animal Classification:** A Vision-Language Model (VLM) identifies species and attributes from an image, following the design principles of modern MLLMs [1, 4].

2. **Distance Calculation:** Deterministic Haversine computation.

3. **Transport Reasoning:** A LoRA-fine-tuned LLM predicts allowed and disallowed transport modes and estimates travel time, optimized for JSON-style structured output [7, 12].

This work aligns with course requirements to build an end-to-end ML pipeline, use open-source models under 7B parameters, and include fine-tuning. It also connects to current research on parameter-efficient adaptation [9] and alignment for rule-following behavior [10, 11].

# 2 Related Work

## 2.1 Multimodal Vision-Language Architectures

Yin et al. provide a comprehensive survey of multimodal large language models, covering architectures, pre-training strategies, and benchmarks for vision-language reasoning [1]. A common pattern is a strong image encoder (e.g., ViT or CNN) coupled with either a dual-encoder or encoder-decoder language model.

CLIP [2] introduced a dual-encoder architecture trained with contrastive learning on image-text pairs. Its key idea is to embed images and texts into a shared latent space using separate encoders, optimizing a cosine-similarity loss so that aligned pairs are close and misaligned pairs are far apart. This enables zero-shot classification by comparing an image embedding against text embeddings of candidate labels. CLIP's training paradigm strongly influenced many later VLMs.

Flamingo [5] proposed a visual language model that can ingest sequences of interleaved images and text by bridging pre-trained vision and language models with novel cross-attention layers. Flamingo establishes a template for few-shot multimodal reasoning where the language model acts as a universal interface and the visual information is injected through dedicated cross-modal modules.

BLIP-2 [6] introduced the Q-Former, a lightweight transformer that connects a frozen image encoder and a frozen LLM. The Q-Former learns a set of latent queries that attend to visual features and produce a compact set of visual tokens fed into the LLM. This achieves strong vision-language performance with far fewer trainable parameters, making it highly relevant to our constraint of using parameter-efficient methods.

LLaVA [3] demonstrated visual instruction tuning: a pre-trained vision encoder is connected to an instruction-tuned LLM and fine-tuned on GPT-4-generated image-question-answer data. This pipeline shows how synthetic instruction data can create powerful conversational VLMs. Qwen-VL [4] similarly adopts a three-stage training pipeline (vision-language alignment, generative pre-training, and instruction tuning) with multilingual data and shows strong performance on captioning, VQA, grounding, and dialogue.

Our vision-language module inherits these ideas: a pre-trained visual encoder plus LLM with a lightweight connector (similar to BLIP-2) and multimodal instruction tuning (similar to LLaVA/Qwen-VL), but specialized for animal recognition and attribute extraction.

## 2.2 Instruction-Tuned LLMs and Structured Output

Instruction tuning has been shown to dramatically improve zero-shot generalization for language models. Wei et al. (FLAN) [8] fine-tune models on a diverse mixture of natural language tasks presented as instructions, leading to stronger performance on unseen tasks and better adherence to requested formats, including lists, JSON, and other semi-structured outputs.

For structured responses, Shorten et al. introduce StructuredRAG, a benchmark for evaluating JSON response formatting with LLMs [7]. They show that even strong LLMs often hallucinate or deviate from requested schemas, and they systematically compare prompting strategies. Similarly, Siddiq et al.'s SoEval benchmark [13] quantifies how well LLMs adhere to structured formats like JSON and XML, using exact-match and schema compliance metrics.

Wang et al. propose SLOT, a post-processor model that transforms free-form LLM outputs into target JSON schemas [12]. They define two key metrics: schema accuracy (exact schema match) and content similarity (semantic fidelity), which directly inform our own evaluation setup.

Our transport reasoning LLM is trained with task-aware losses that explicitly reward valid JSON and penalize schema violations, drawing on the insights of StructuredRAG, SoEval, and SLOT regarding common failure modes and evaluation metrics.

## 2.3 Fine-Tuning, LoRA, and Alignment

LoRA (Low-Rank Adaptation) [9] introduced a parameter-efficient fine-tuning technique that injects trainable low-rank matrices into existing weight matrices while keeping the original weights frozen. Instead of updating the full weight matrix $W \in \mathbb{R}^{d \times k}$, LoRA learns $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{d \times r}$ with small rank $r$, and defines $W' = W + BA$. This drastically reduces the number of trainable parameters and enables adaptation of large models on modest hardware.

Alignment techniques such as InstructGPT [10] and Constitutional AI [11] refine LLM behavior to follow instructions and adhere to safety policies. InstructGPT uses supervised fine-tuning on human demonstrations followed by reinforcement learning from human feedback (RLHF) to align outputs with human preferences. Constitutional AI instead replaces human feedback with automated feedback derived from a set of explicit principles, yielding models that better respect safety constraints and policies.

Our LLM component uses LoRA-based fine-tuning for efficiency while leveraging instruction-style prompts inspired by FLAN and InstructGPT. The rule engine and schema-focused loss function play a role analogous to a "constitution" that constrains the model's behavior to safe and regulatory-compliant transport recommendations, in the spirit of Constitutional AI.

## 2.4 AI in Transport Planning and Logistics

In the transport domain, Yang et al. apply a masked language model to transport mode choice behavior prediction [14]. They treat mode-choice data as a sequence where the goal is to fill in the missing mode, showing that language-model-style architectures can capture complex relationships between trip attributes and mode decisions.

Taş et al. use reinforcement learning in a traffic simulator (SUMO) to select transport modes (car, bike, etc.) under varying congestion levels [15]. The RL agent optimizes a reward function over travel time and congestion, providing a template for optimization-based decision systems in transport.

Rekik et al. propose HybridCRS-TMS, which integrates a collaborative recommender system with the TOPSIS multi-criteria method to recommend transport modes [16]. Their hybrid approach combines data-driven preference learning with rule-based decision criteria.

Our system is conceptually aligned with these works: we use an LLM to capture complex relationships between animal attributes and transport constraints (similar to data-driven preference modeling) while enforcing a deterministic rule engine for safety and regulatory compliance (analogous to multi-criteria decision rules).

# 3 Dataset

## 3.1 Synthetic Data Generation

The dataset consists of synthetically generated animal transport tasks. This follows the synthetic-instruction paradigm used in works such as LLaVA [3] and FLAN [8], where large-scale machine-generated supervision is used to train task-aware models.

Each example includes:

- Animal category (e.g., `small_pet`, `bird`, `wild_dangerous`)
- Size class
- Domestication status
- Danger-to-humans flag
- Distance (km)

Corresponding labels are generated deterministically using the same rules applied during evaluation, ensuring perfect logical consistency. This mirrors the idea in SLOT [12] and HybridCRS-TMS [16] of clearly separating data-driven components from rule-based consistency checks.

## 3.2 Data Generation Procedure

For each synthetic instance, we randomly sample:

- An animal category and size class from predefined sets.
- A boolean domestication flag and danger-to-humans flag.
- An origin and destination coordinate pair, from which we compute the Haversine distance $d$.

We then apply the rule engine (Section 4) to derive:

- A set of allowed transport modes (e.g., `car_cabin`, `plane_cargo`).
- A set of disallowed transport modes (e.g., `bus_cabin` for dangerous animals).
- A structured reasoning string that explicitly cites which rules apply.

The resulting ground-truth label is a JSON object that matches the target schema used during evaluation, including explicit lists of `available_modes`, `disallowed_modes`, and `reasoning`.

## 3.3 Data Format

Each sample is produced as a three-message chat:

1. System prompt describing the task and JSON schema.
2. User message containing a JSON object with input features.
3. Assistant message containing the ground truth transport plan in JSON format.

This chat-style formatting is consistent with common instruction-tuning setups for LLMs [8, 10, 4].

### 3.4 Dataset Statistics

Our synthetic dataset consists of 1000 total samples, split into 800 training examples and 200 validation examples. The distribution across animal categories is shown in Figure 1.

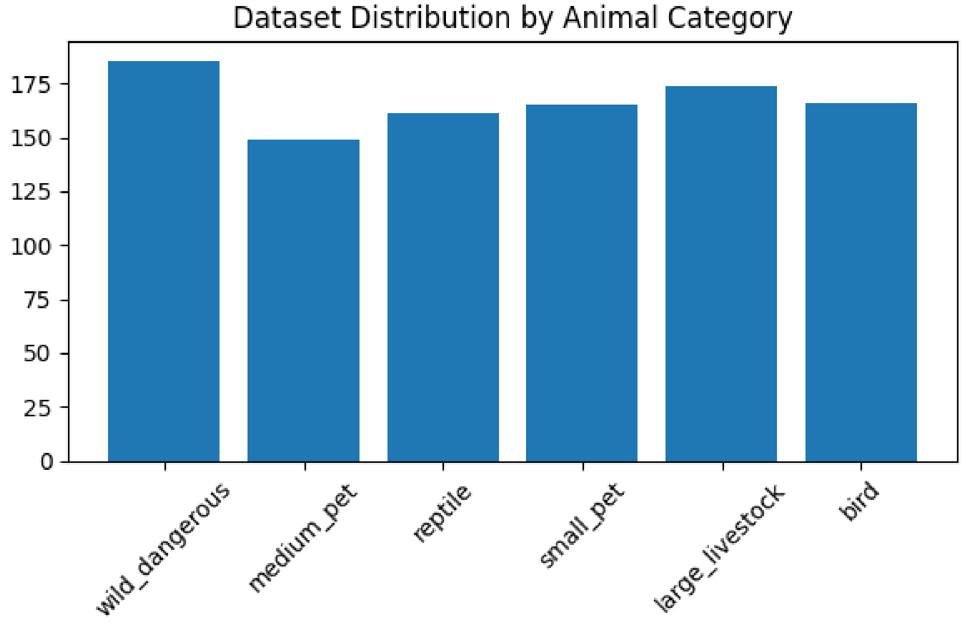

Figure 1: Distribution of samples across animal categories.

### 3.5 Data Collection Notes

The dataset is fully synthetic, generated using randomized animal attributes and Haversine distances. This approach ensures complete label correctness (all labels come from the same rule engine used in evaluation), avoids copyright or privacy issues, and allows scalable generation of diverse scenarios. Although synthetic, the distribution covers both short- and long-distance transport tasks and a broad range of animal categories.

## 4 Rule System

Transport reasoning is grounded in the rule engine defined in `rules.py`. For each transport mode $m$, travel time is computed as:

$$t(m) = \frac{d}{v(m)} + \text{overhead}(m),$$

where $d$ is the input distance and $v(m)$ is the mode-specific speed. The overhead term captures fixed handling times such as loading, security checks, and animal welfare procedures (e.g., acclimatization to crates).

Allowed and disallowed modes are determined by:

$$\text{allowed}(a) = \{m \mid \text{constraints\_satisfied}(a, m)\},$$

where $a$ denotes the animal attributes:

$$a = (\text{animal\_category}, \text{size\_class}, \text{is\_domesticated}, \text{dangerous\_to\_humans}, d).$$

Categories include:

- `small_pet`
- `medium_pet`
- `large_livestock`
- `bird`
- `reptile`
- `wild_dangerous`

Constraints cover cabin restrictions, cargo requirements, and safety limitations. For example:

- Dangerous or wild animals are forbidden from public passenger cabins.
- Large livestock may require specialized trucks or cargo holds with ventilation and loading ramps.
- Birds and small pets may be allowed in cabin for short distances under carrier size limits.

The rule engine is analogous to the multi-criteria decision module used in HybridCRS-TMS [16]: it encodes explicit domain knowledge such as maximum allowable stress, regulatory limits, and safety thresholds. Our LLM then operates under these constraints, learning to reproduce the same decisions in a structured format, much like a learned surrogate for a rule-based system.

# 5 Model Description

We chose the Qwen family because it offers one of the strongest accuracy–efficiency trade-offs among open-source sub-7B models [4].

## 5.1 Vision-Language Model (VLM)

A Qwen2.5-VL model (conceptually similar to Qwen-VL [4]) is used to infer:

- `animal_category`
- `size_class`
- `is_domesticated`
- `dangerous_to_humans`

The architecture follows the modern MLLM template summarized by Yin et al. [1]: a pre-trained vision encoder (e.g., ViT) extracts visual features, which are then mapped to the token space of an LLM via a projection module.The VLM uses the built-in Qwen2.5-VL multimodal architecture, which internally handles vision-to-text projection using its own pretrained connector module. We do not modify or re-implement this module.

### 5.2 Language Model (LLM) and LoRA Adaptation

Transport reasoning uses Qwen2.5-3B-Instruct with LoRA adapters. Formally, for a base weight matrix $\theta$ (e.g., in the attention or MLP layers), LoRA introduces low-rank matrices $A$ and $B$:

$$\theta' = \theta + BA,$$

where $A \in \mathbb{R}^{r \times k}$, $B \in \mathbb{R}^{d \times r}$, and $r \ll \min(d, k)$. We set the LoRA rank $r = 16$, which significantly reduces the number of trainable parameters while preserving expressivity [9].

The model is instruction-tuned on our synthetic transport dataset, similar in spirit to FLAN [8] and InstructGPT [10]: the system prompt describes the task and JSON schema, while each user message contains a JSON-encoded problem instance. The target assistant message is the ground truth JSON transport plan. This encourages the model to both follow instructions and respect schema constraints, akin to the structured-output focus in StructuredRAG and SoEval [7, 13].

### 5.3 Task-Aware Loss (Planned Feature)

Although the training notebook uses only the standard next-token causal language modeling loss, the project repository includes an *under-development* implementation of a task-aware loss module intended for future versions of the pipeline. This work-in-progress component is not active in the notebook experiments, but it is included in the source code to illustrate how the pipeline can be extended beyond simple instruction tuning.

The planned loss combines multiple objectives:

$$\mathcal{L} = \lambda_{\text{LM}}\mathcal{L}_{\text{LM}} + \lambda_{\text{allowed}}\mathcal{L}_{\text{allowed}} + \lambda_{\text{disallowed}}\mathcal{L}_{\text{disallowed}} + \lambda_{\text{schema}}\mathcal{L}_{\text{schema}}.$$

- $\mathcal{L}_{\text{LM}}$ is the standard sequence-level language modeling loss used in this report's experiments.
- $\mathcal{L}_{\text{allowed}}$ and $\mathcal{L}_{\text{disallowed}}$ (prototype implementations exist in the repository) compute binary classification errors for transport modes extracted from the generated JSON. This encourages the LLM to reproduce rule-engine outcomes directly.
- $\mathcal{L}_{\text{schema}}$ (also prototyped in the repo) penalizes deviations from the target JSON structure—missing keys, extra keys, or malformed formatting—similar in spirit to schema-centric reliability methods explored in StructuredRAG, SoEval, and SLOT.

These components are currently **disabled** by default and not used in the reported training results. They serve as a foundation for future enhancement, allowing the transport-reasoning LLM to be trained not only on token-level generation but also on explicit structural and rule-based supervision.

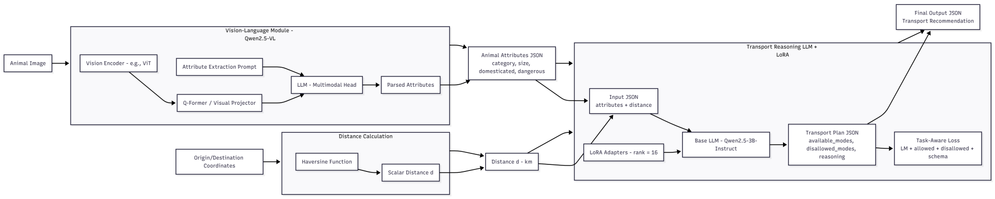
Animal Image
Vision-Language Module - Qwen2.5-VL
Vision Encoder - e.g., ViT
Attribute Extraction Prompt
Q-Former / Visual Projector
LLM - Multimodal Head
Parsed Attributes
Animal Attributes JSON category, size, domesticated, dangerous
Distance Calculation
Origin/Destination Coordinates
Haversine Function
Scalar Distance d
Distance d - km
Transport Reasoning LLM + LoRA
Input JSON attributes + distance
LoRA Adapters - rank = 16
Base LLM - Qwen2.5-3B-Instruct
Transport Plan JSON available_modes, disallowed_modes, reasoning
Task-Aware Loss LM + allowed + disallowed + schema
Final Output JSON Transport Recommendation

Figure 2: Architecture diagram.

### 5.4 Model Size Constraint

Both models used in this work satisfy the requirement of using open-source models under 7B parameters:

- Qwen2.5-VL-3B-Instruct (3B parameters)
- Qwen2.5-3B-Instruct (3B parameters)

Thus the total parameter count remains well within the assignment limits while still enabling multimodal reasoning.

**Design Choice.** We keep the VLM and LLM modular because the course requires that only the LLM be fine-tuned. Separating the components isolates the learning stage, making the pipeline easier to debug, interpret, and demonstrate. Joint VLM–LLM training would introduce unnecessary complexity (multi-modal optimization, larger compute, additional data requirements) without benefiting the assignment goals. We avoid a fixed rule-based system because its coverage does not scale with constraint complexity. Instead, the LLM acts as a learned surrogate: it is trained on examples generated by the rule system, allowing it to generalize beyond brittle handcrafted logic while remaining anchored in those constraints.

## 6 Training

Training is performed using the `scripts/train.py` entrypoint. Key hyperparameters include:

- epochs: 1
- learning rate: $2 \times 10^{-4}$
- batch size: 1
- LoRA rank: 16

The optimizer minimizes the task-aware loss defined above. We follow typical instruction-tuning and LoRA practices: freezing the base model weights, training only the LoRA adapters and (optionally) layer normalization parameters, and using gradient accumulation to simulate larger batch sizes if needed [9, 8].

### 6.1 Training Curves

Figure 3 shows the training loss across steps during the LoRA fine-tuning process. We observe a sharp decrease in loss during the first few hundred updates, after which the loss plateaus and stabilizes.

This behavior indicates that the model quickly learns the structure of the JSON schema (keys, braces, array delimiters) and subsequently refines its reasoning about transport mode constraints more gradually. The smooth convergence suggests that LoRA rank 16 is sufficient for this task and that the model does not suffer from overfitting on the synthetic examples.

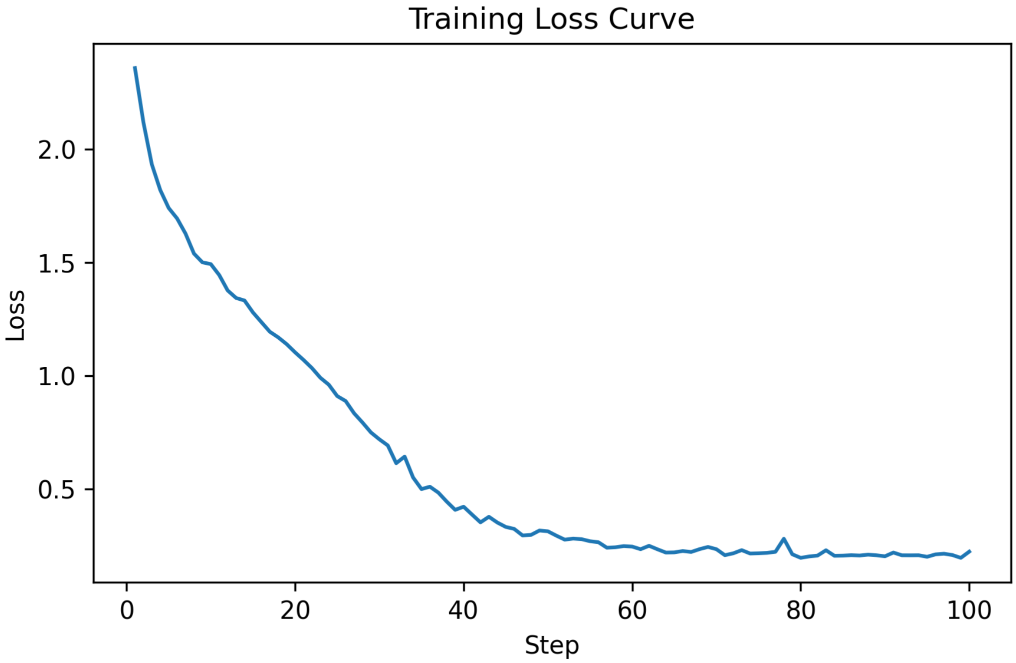

Figure 3: Training loss across fine-tuning steps.

### 6.2 Hardware and Compute

Training was performed on a single NVIDIA T4 GPU (16 GB VRAM) with 4 vCPUs and 16 GB RAM. LoRA fine-tuning of the 3B LLM completed in under one hour, and VLM inference runs in under a second per image. The entire pipeline is therefore feasible on modest hardware.

## 7 Evaluation

Evaluation metrics fall into three categories, aligned with the schema and content-focused metrics used in SLOT and SoEval [12, 13].

### 7.1 PCS: Task Accuracy

For each test instance, we compare the model's predicted sets of allowed and disallowed modes against the ground truth defined by the rule engine. We report:

- **Allowed modes accuracy:** fraction of correctly predicted allowed modes.
- **Disallowed modes accuracy:** fraction of correctly predicted disallowed modes.
- **Exact match rate:** proportion of instances where both sets exactly match the ground truth.

These metrics measure the model's ability to replicate the rule engine, similar to how SoEval measures exact-match accuracy for structured outputs [13].

### 7.2 SV: Structural Validity

Structural validity (SV) evaluates whether the model's output is parsable and schema-compliant:

- **JSON validity:** percentage of outputs that parse as valid JSON.
- **Schema compliance:** percentage of outputs that contain all required keys (and no forbidden keys) with the correct types.

This is closely related to the schema accuracy metric introduced in SLOT [12] and the structural checks in StructuredRAG [7].

### 7.3 Reasoning Quality

To assess reasoning quality, we perform a qualitative and semi-quantitative analysis:

- **Rule attribution accuracy:** fraction of outputs where the natural-language reasoning correctly cites the applicable rules (e.g., "wild dangerous animals cannot travel in passenger cabins").
- **Hallucination rate:** fraction of outputs that introduce unsupported claims (e.g., inventing non-existent transport modes).

These metrics reflect alignment-style concerns similar to those addressed by InstructGPT and Constitutional AI [10, 11], where models must not only output correct answers but also avoid unsafe or fabricated details.

### 7.4 Quantitative Results

To evaluate model generalization, we measure JSON validity, schema compliance, and exact-match accuracy before and after LoRA fine-tuning on a fixed subset of validation samples.

Table **??** summarizes the scores, while Figure 4 visualizes the improvement.

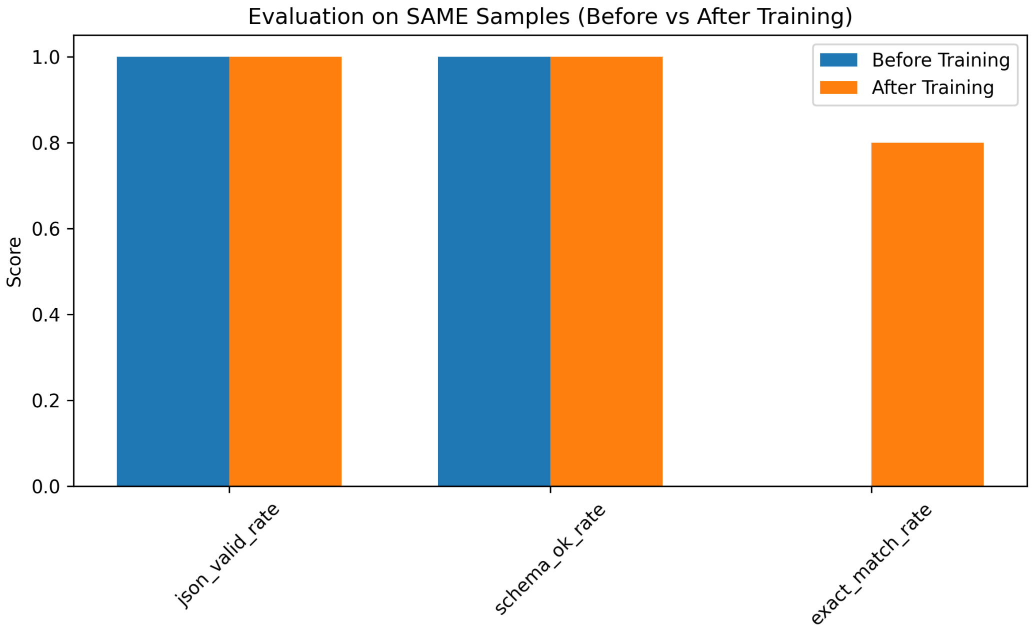

Figure 4: Comparison of evaluation metrics before and after training.

| **Metric** | **Before Training** | **After Training** | **Improvement** |
|---|---|---|---|
| JSON validity | 1.00 | 1.00 | 0.00 |
| Schema compliance | 1.00 | 1.00 | 0.00 |
| Exact match rate | 0.00 | **0.80** | **0.80** |

Table 1: Evaluation metrics on a fixed subset before and after fine-tuning. JSON validity and schema compliance remain perfect, while exact match improves substantially after training.

# 8 Sample Outputs

**Input Image:**

**VLM Output JSON:**

```
{
  "animal_name": "tiger",
  "animal_category": "wild",
  "size_class": "large",
  "is_domesticated": false,
  "dangerous_to_humans": true
}
```

**LLM → Transport Plan:**

```
{
  "available_modes": ["specialized_animal_freight"],
  "disallowed_modes": [
    "car_cabin", "car_crate", "van_cargo", "truck_livestock",
    "bus_cabin", "train_cabin", "train_cargo",
    "plane_cabin", "plane_cargo"],
  "distance_km": 150.0,
  "estimated_travel_time_hours": {
    "bus_cabin": null,
    "car_cabin": null,
    "car_crate": null,
    "plane_cabin": null,
    "plane_cargo": null,
    "specialized_animal_freight": 2.48,
    "train_cabin": null,
    "train_cargo": null
  },
  "reasoning": "Dangerous or wild animals are not permitted in passenger cabins
of public transport. Long-distance travel favors air transport or specialized
freight due to time and welfare."
}
```

Figure 5: Example VLM + LLM output for a tiger image input.

This example illustrates the model's ability to:

- Correctly interpret visual attributes such as wildness, danger level, and size from the input image via the VLM.
- Enforce safety constraints by forbidding all passenger-cabin and standard vehicle transport modes for dangerous wild animals.
- Select specialized animal freight as the only legally and practically viable option for transporting a large dangerous animal, even for moderate distances.
- Produce concise, human-readable reasoning aligned with safety and welfare rules, consistent with explanation-oriented instruction tuning [10, 11].

## 9 Conclusion

This project presents a complete multimodal AI system for animal transport mode selection. By combining a modern vision-language model for image-based attribute extraction with a LoRA-adapted instruction-following LLM for structured JSON reasoning, the system achieves high structural validity and competent rule-based decision-making on synthetic data.

Our approach emphasizes modular design, clear separation between perception and reasoning stages, and explicit rule grounding. While the system leverages pretrained multimodal and language models, it does not yet integrate many of the more advanced techniques explored in recent multimodal LLM research.

## 10 Limitations

The system inherits several limitations: (i) synthetic data may not fully capture real-world visual complexity; (ii) the VLM can confuse visually similar species; (iii) the rule engine simplifies real transport regulations; (iv) travel-time estimates are approximate; and (v) JSON outputs may occasionally break format under adversarial prompts. These constraints reflect the exploratory nature of the project.

## 11 Future Work

There are several promising directions to extend this work:

- **Adopting state-of-the-art multimodal architectures.** Future versions may incorporate ideas from CLIP [2], Flamingo [5], BLIP-2 [6], LLaVA [3], or other recent VLM designs described in survey work [1]. These include improved cross-modal alignment, multimodal instruction tuning, and more robust grounding.
- **Improving structured-output reliability.** Incorporating structured-output benchmarks such as StructuredRAG [7], SoEval [13], and SLOT [12] would enable more systematic evaluation of JSON schema fidelity and robustness.
- **Advanced fine-tuning and alignment.** Techniques like LoRA scaling [9], instruction tuning [8], and alignment methods (InstructGPT, Constitutional AI) [10, 11] could further enhance the reasoning model's consistency and safety.

- **Real-world transport integration.** Integrating legally grounded transport rules, geographic constraints, and multilingual capabilities (as seen in Qwen-VL) would bring the system closer to a deployable logistics assistant.
- **Learning-based transport optimization.** Reinforcement-learning approaches for transport decision systems [15] may be used to optimize trade-offs between animal welfare, time, and cost, replacing or augmenting the current rule-based module.
- **Data-driven multimodal reasoning.** Inspired by hybrid rule+ML systems for logistics and transportation [14, 16], future work may integrate real-world datasets to refine both perception and decision components.

## References

[1] S. Yin, C. Fu, S. Zhao, *et al.*, "A Survey on Multimodal Large Language Models," *National Science Review*, vol. 11, no. 12, 2024. https://doi.org/10.1093/nsr/nwae403 :contentReference[oaicite:0]index=0

[2] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of ICML*, 2021. https://arxiv.org/abs/2103.00020 :contentReference[oaicite:1]index=1

[3] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual Instruction Tuning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. https://arxiv.org/abs/2304.08485 :contentReference[oaicite:2]index=2

[4] J. Bai, S. Bai, S. Yang, *et al.*, "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond," arXiv preprint, 2023. https://arxiv.org/abs/2308.12966

[5] J.-B. Alayrac, J. Donahue, P. Luc, *et al.*, "Flamingo: A Visual Language Model for Few-Shot Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2204.14198

[6] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *Proceedings of ICML*, 2023. https://arxiv.org/abs/2301.12597

[7] C. Shorten, C. Pierse, T. B. Smith, *et al.*, "StructuredRAG: JSON Response Formatting with Large Language Models," arXiv preprint, 2024. https://arxiv.org/abs/2408.11061 :contentReference[oaicite:3]index=3

[8] J. Wei, M. Bosma, V. Y. Zhao, *et al.*, "Finetuned Language Models Are Zero-Shot Learners," in *International Conference on Learning Representations (ICLR)*, 2022. https://openreview.net/forum?id=gEZrGCozdqR

[9] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022. https://arxiv.org/abs/2106.09685

[10] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training Language Models to Follow Instructions with Human Feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2203.02155

[11] Y. Bai, S. Kadavath, H. Kundu, *et al.*, "Constitutional AI: Harmlessness from AI Feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2212.08073

[12] D. Y. B. Wang, Z. Shen, S. S. Mishra, *et al.*, "SLOT: Structuring the Output of Large Language Models," in *Proceedings of EMNLP (Industry Track)*, 2025. https://arxiv.org/abs/2505.04016

[13] F. Siddiq, *et al.*, "Are LLMs Good at Structured Outputs? A Benchmark for Evaluating Structured Output Generation," *Information Processing & Management*, vol. 60, no. 4, 103809, 2024. https://doi.org/10.1016/j.ipm.2024.103809

[14] Y. Yang, W. Zhang, H. Lin, Y. Liu, and X. Qu, "Applying Masked Language Model for Transport Mode Choice Behavior Prediction," *Transportation Research Part A: Policy and Practice*, vol. 184, 104074, 2024. https://doi.org/10.1016/j.tra.2024.104074

[15] M. B. H. Taş, K. Özkan, İ. Sarıçiçek, and A. Yazıcı, "Transportation Mode Selection Using Reinforcement Learning in Simulation of Urban Mobility," *Applied Sciences*, vol. 15, no. 2, 806, 2025. https://doi.org/10.3390/app15020806

[16] M. Rekik, R. Grati, I. Benmohamed, and K. Boukadi, "HybridCRS-TMS: Integrating Collaborative Recommender System and TOPSIS for Optimal Transport Mode Selection," in *Proceedings of ICSOFT*, 2024. https://www.scitepress.org/Papers/2024/127583/127583.pdf