CausalVAD: De-confounding End-to-End Autonomous Driving via Causal Intervention

Supplementary Material

This document serves as a substantial extension to the main manuscript, providing a theoretical grounding for the proposed sparse causal intervention scheme (SCIS) and detailing its implementation, efficiency, and empirical validation. We structure this material to ensure theoretical depth and reproducibility. **Section 1** presents a derivation linking Pearl's backdoor adjustment to our neural subtraction mechanism and theoretically distinguishes our method from heuristic baselines. **Section 2** details the tensor-level formulations of the perception de-confounding module (PDM) and interaction de-confounding module (IDM). **Section 3** elaborates on the optimization strategy, specifically addressing the rationale behind the offline dictionary construction. Finally, **Section 4** provides comprehensive empirical evidence, including baseline reproduction details, counterfactual perturbation studies, zero-shot cross-domain and cross-paradigm generalization, hyperparameter sensitivity, upstream task performance, computational complexity analysis, physical validity stress tests, and qualitative interpretability.

1. Theoretical grounding of SCIS

The core theoretical challenge addressed in this work is the translation of causal intervention, formally defined in the probability space, into differentiable operations within a deep neural network.

1.1. From probabilistic adjustment to feature space

Consider a standard structural causal model (SCM) where the input variable S, the target prediction Y, and the latent confounder Z form a classic confounding structure: $S \leftarrow Z \rightarrow Y$. Standard empirical risk minimization fits the observational distribution $P(Y|S)$, capturing spurious correlations induced by Z. To learn the true causal effect, we attempt to estimate the interventional distribution $P(Y|\text{do}(S))$. According to the backdoor adjustment formula [3], this is given by:

$$P(Y|\text{do}(S)) = \sum_{z \in \mathcal{Z}} P(Y|S,z)P(z) \tag{S.1}$$

Directly computing this integral is intractable in high-dimensional continuous latent spaces. Therefore, our formulation should be interpreted as a tractable *approximation of backdoor adjustment*, rather than a rigorous mathematical proof of causal identifiability. This approximation provides a pragmatic operationalization of causal theory for large-scale end-to-end autonomous driving models. To proceed, we introduce two critical assumptions rooted in the analysis of deep representations.

Assumption 1: the additive bias hypothesis. We posit that in the high-dimensional feature space (or logit space) of a trained neural network, the representation of a confounded sample can be decomposed into a causal component and a spurious component. Formally, let $\phi(S,z)$ denote the feature representation (or logits) given input S and context z. We assume:

$$\phi(S,z) \approx \phi_{causal}(S) + \lambda \cdot \phi_{spur}(z) \tag{S.2}$$

where $\phi_{causal}(S)$ represents the intrinsic features of the object, and $\phi_{spur}(z)$ represents the features induced by the environmental context.

Assumption 2: NWGM approximation. Following the normalized weighted geometric mean (NWGM) theory [6, 7], the expectation of a variable inside a softmax activation can be approximated by the softmax of the expectation. This allows us to move the summation operation from Equation (S.1) into the logit space.

Combining these assumptions, the interventional prediction can be approximated as:

$$P(Y|\text{do}(S)) \approx \text{Softmax}\left(\phi_{causal}(S) + \mathbb{E}_z[\phi_{spur}(z)]\right) \tag{S.3}$$

However, a standard model trained on biased data learns to output the entangled representation $\phi_{obs} \approx \phi_{causal}(S) + \mathbb{E}_z[\phi_{spur}(z)]$. Consequently, to recover the true causal component $\phi_{causal}(S)$, we perform a subtractive operation:

$$\phi_{causal}(S) \approx \phi_{obs} - \mathbb{E}_z[\phi_{spur}(z)] \tag{S.4}$$

This derivation provides the theoretical legitimacy for our design: we estimate the expectation of the confounder bias $\mathbb{E}_z[\phi_{spur}(z)]$ using a dictionary and subtract it from the network's representations.

1.2. Intervention at different SCM nodes

Our framework applies this subtractive intervention at two distinct loci: the output logits (PDM) and the latent input features (IDM). These are theoretically consistent within the SCM framework.

The perception de-confounding module (PDM) operates on the output variable Y (specifically, its logits). Since the mapping from the final layer features to logits is linear, subtracting the bias in the logit space is mathematically equivalent to re-weighting the class posterior probabilities to remove the prior $P(Y|Z)$.

The interaction de-confounding module (IDM) operates on the input variable S of the interaction blocks. By intervening on the query tensor Q *before* interaction (i.e., $Q_{new} \leftarrow Q - \text{Bias}$), we effectively perform a latent variable intervention. This blocks the flow of information from Z to the downstream interaction module, ensuring that the subsequent modeling relies solely on causal features.

1.3. Theoretical superiority over heuristic baselines

A pertinent question is whether simple regularization techniques could achieve similar robustness. We argue that heuristics like dropout [2] or data augmentation [1] are fundamentally insufficient compared to SCIS.

Dropout, even when targeted at specific states (e.g., dropout on ego status) [2], acts as an *indiscriminate* regularization. It randomly suppresses features with equal probability, potentially discarding valid causal information (e.g., valid velocity cues) along with spurious ones. In contrast, SCIS performs a *surgical* intervention: it only subtracts components that semantically align with the discovered confounder prototypes in the dictionary, thereby preserving valid causal dynamics. Similarly, while data augmentation [1] attempts to intervene on the input pixel space (e.g., changing weather), it is limited by the realism and diversity of the simulator. SCIS can be viewed as a feature-level causal augmentation that operates directly on the latent manifold, offering a more dense and effective coverage of the confounder space than pixel-level manipulation.

2. Formal implementation details

We hereby present the precise mathematical formulation of the proposed modules using unified tensor notation.

2.1. Perception de-confounding module (PDM)

The PDM targets classification tasks where the final logits are susceptible to co-occurrence bias. Let $\mathbf{Q} \in \mathbb{R}^{B\times N\times D}$ denote a batch of query features (e.g., map queries), and $\mathbf{L}_{obs} \in \mathbb{R}^{B\times N\times C}$ denote the observational logits produced by the classifier head. We utilize a pre-computed confounder dictionary $\mathbf{Z} \in \mathbb{R}^{K\times D}$.

To estimate the bias specific to each query, we first compute the affinity matrix $\mathbf{A} \in \mathbb{R}^{B\times N\times K}$ via dot-product attention:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{Z}^\top}{\sqrt{D}}\right) \tag{S.5}$$

This attention mechanism retrieves the distribution of context prototypes relevant to the current query. We then project these prototypes into the logit space using a learnable projection matrix $\mathbf{B}_{proto} \in \mathbb{R}^{K\times C}$. The estimated logit bias is:

$$\mathbf{L}_{bias} = \mathbf{A}\mathbf{B}_{proto} \tag{S.6}$$

Finally, we perform the causal intervention via adaptive subtraction:

$$\mathbf{L}_{final} = \mathbf{L}_{obs} - \boldsymbol{\lambda} \otimes \mathbf{L}_{bias} \tag{S.7}$$

Crucially, $\boldsymbol{\lambda} \in \mathbb{R}^C$ is defined as a learnable parameter vector (broadcastable across batch and queries). This design choice is significant: it allows the network to automatically learn the magnitude of bias associated with each specific class. For instance, *pedestrians* might suffer from heavier contextual bias than *cars*, and a learnable vector $\boldsymbol{\lambda}$ can capture this heterogeneity.

2.2. Interaction de-confounding module (IDM)

The IDM is designed to remove spurious components from intermediate feature tensors before they participate in cross-attention mechanisms (e.g., Ego query attending to Agent features). Let $\mathbf{S}_{in} \in \mathbb{R}^{B\times N\times D}$ be the input tensor suspected of confounding.

We treat the estimation of the spurious component as a reconstruction problem using multi-head cross-attention (MHCA). The input $\mathbf{S}_{in}$ serves as the Query, while the confounder dictionary $\mathbf{Z}$ serves as both Key and Value:

$$\mathbf{C}_{spur} = \text{MHCA}(\text{Q} = \mathbf{S}_{in}, \text{K} = \mathbf{Z}, \text{V} = \mathbf{Z}) \tag{S.8}$$

Note that we do not treat the BEV representation itself as a confounder. Instead, we target the entangled statistical priors within BEV interactions (e.g., map topology dominating agent prediction due to inertia bias). By refining these features, the IDM removes this task-irrelevant correlation while preserving physical mediator signals (as subsequently visualized in Figure S.1).

Furthermore, to prevent *over-deconfounding*, we employ a gating mechanism. A non-linear perceptron computes a gating tensor $\mathbf{G} \in \mathbb{R}^{B\times N\times D}$ based on the concatenation of the original and spurious features:

$$\mathbf{G} = \sigma\left(\text{MLP}\left(\text{Concat}(\mathbf{S}_{in}, \mathbf{C}_{spur})\right)\right) \tag{S.9}$$

This learnable gating unit dynamically adjusts the intervention intensity, preserving essential causal signals while pruning spurious ones. The refined feature tensor $\mathbf{S}_{clean}$ is obtained by subtracting the gated spurious signal:

$$\mathbf{S}_{clean} = \mathbf{S}_{in} - \mathbf{G} \otimes \mathbf{C}_{spur} \tag{S.10}$$

This refined tensor $\mathbf{S}_{clean}$ is then used as the clean input for subsequent transformer layers, effectively severing the latent backdoor path.

3. Training strategy and optimization

To ensure the stability of the causal intervention, we adopt a two-stage training paradigm. We summarize the complete pipeline in Algorithm 1.

Algorithm 1 CausalVAD training pipeline

Require: dataset $\mathcal{D}$, pre-trained VAD model Φ_{pre}
Ensure: CausalVAD model parameters θ
// Stage 1: dictionary construction (offline)
Initialize empty storage buffers $\mathbb{S}_o, \mathbb{S}_m, \mathbb{S}_a$.
for each batch $(\mathbf{X}, \mathbf{Y}_{gt})$ in $\mathcal{D}$ **do**
  Extract latent queries using frozen Φ_{pre}:
  $Q_o, Q_m, Q_a \leftarrow \Phi_{pre}(\mathbf{X})$
  Append queries to buffers: $\mathbb{S}_o \leftarrow Q_o, \mathbb{S}_m \leftarrow Q_m, \mathbb{S}_a \leftarrow Q_a$
end for
Perform clustering to obtain prototypes:
$\mathcal{Z}_o \leftarrow$ K-Means++$(\mathbb{S}_o, K = k_o)$
$\mathcal{Z}_m \leftarrow$ K-Means++$(\mathbb{S}_m, K = k_m)$
$\mathcal{Z}_a \leftarrow$ K-Means++$(\mathbb{S}_a, K = k_a)$
// Stage 2: causal intervention training (online)
Initialize θ for CausalVAD. Load $\mathcal{Z} = \{\mathcal{Z}_o, \mathcal{Z}_m, \mathcal{Z}_a\}$ as constants.
while not converged **do**
  Sample batch $(\mathbf{X}, \mathbf{Y}_{gt})$ from $\mathcal{D}$.
  Extract BEV features $\mathbf{B}$.
  Initialize queries Q_o, Q_m, Q_a, Q_e.
  // Step 2.1: perception stage (PDM on logits)
  $Q_o \leftarrow$ DetectionDecoder$(Q_o, \mathbf{B})$
  $Q_m \leftarrow$ MappingDecoder$(Q_m, \mathbf{B})$
  $L'_o \leftarrow$ PDM$(Q_o, \mathcal{Z}_o)$ ▷ De-confound object det.
  $L'_m \leftarrow$ PDM$(Q_m, \mathcal{Z}_m)$ ▷ De-confound map seg.
  // Step 2.2: prediction stage (IDM on features)
  $Q'_o \leftarrow$ IDM$(Q_o, \mathcal{Z}_m)$ ▷ De-confound object using map context
  $Q'_m \leftarrow$ IDM$(Q_m, \mathcal{Z}_o)$ ▷ De-confound map using obj. context
  $Q_a \leftarrow$ MotionDecoder(Q_a, Q'_o, Q'_m)
  // Step 2.3: planning stage (IDM on features)
  $Q'_a \leftarrow$ IDM$(Q_a, \mathcal{Z}_m)$ ▷ De-confound agent using map context
  $Q''_m \leftarrow$ IDM$(Q_m, \mathcal{Z}_a)$ ▷ De-confound map using agent context
  $Q_e \leftarrow$ PlanningDecoder(Q_e, Q'_a, Q''_m)
  // Step 2.4: optimization
  Predict trajectories $\mathbf{Y}_{pred}$ from Q_e.
  $\mathcal{L} \leftarrow$ AggregateLoss$(\mathcal{L}_{det}, \mathcal{L}_{map}, \mathcal{L}_{mot}, \mathcal{L}_{plan})$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end while

A critical design choice in our framework is the use of an offline, frozen confounder dictionary during the second stage of training. We argue that updating the dictionary online with the model parameters would lead to a *representation collapse* or adaptation risk. This deliberate design is grounded in the concept of *semantic anchoring*. Although feature representations evolve continuously during end-to-end training, their semantic topology is anchored by shared supervision, allowing the frozen dictionary $\mathcal{Z}$ to capture the intrinsic directions of spurious correlations within the dataset manifold.

By treating $\mathcal{Z}$ as a *negative anchor*, we impose an *orthogonalization constraint*. Specifically, if the dictionary $\mathcal{Z}$ were learnable during the de-confounding phase, the optimizer could minimize the loss by simply pushing the dictionary prototypes towards zero or aligning them with the causal features, thereby trivializing the subtractive penalty. Freezing the dictionary forces the gradients to explicitly drive the newly generated queries $\mathbf{S}$ to diverge from the spurious subspace, effectively severing the spurious links and achieving genuine de-confounding.

Table S.1. Performance on perception and motion prediction tasks.

Method	Detection		Mapping	Motion		
	NDS ↑	mAP ↑	mAP ↑	minADE ↓	minFDE ↓	MR ↓
VAD-tiny	40.73	27.37	48.62	0.87	1.20	0.15
Ours	42.42	30.46	50.97	0.72	0.96	0.10

Table S.2. Computational analysis on one NVIDIA RTX 3090 GPU.

Method	Params (M)	Latency (ms)	FPS
VAD-tiny	119.3	179	5.6
Ours	124.0	185	5.4

4. Extended experimental analysis

4.1. Hyperparameter rationale

In the main manuscript, we reported the optimal dictionary sizes (k_o, k_m, k_a) as $(10, 3, 6)$. This selection is not arbitrary but is guided by domain knowledge regarding the semantic granularity of the nuScenes dataset. Specifically, the nuScenes detection task involves 10 object categories; thus, setting $k_o = 10$ allows the dictionary to capture at least one dominant confounding context prototype per object class. Similarly, the map segmentation task involves 3 primary classes (lane divider, pedestrian crossing, road boundary), justifying $k_m = 3$. For agent motion, the multimodal prediction typically outputs 6 trajectory modes, suggesting that the latent intent space can be well-represented by $k_a = 6$ prototypes. This semantic alignment explains why this specific configuration outperforms others in our ablation studies, as it matches the intrinsic dimensionality of the task-specific confounding factors.

4.2. Impact on upstream tasks

A primary concern with subtractive intervention is whether it degrades the fundamental perception capabilities. We report the performance on upstream tasks in Table S.1. The

Table S.3. Interaction-specific perturbation for counterfactual test. The noise denotes the magnitude of uniform perturbations applied to context features.

Method	Context Noise	Ego-Agent Int.		Ego-Map Int.	
		Avg. L2 (m) ↓	Avg. CR (%) ↓	Avg. L2 (m) ↓	Avg. CR (%) ↓
VAD-tiny	-	0.74	0.44	0.74	0.44
	$\times 0.5$	0.79	0.65	0.79	0.44
	$\times 0.7$	0.84	1.77	1.08	0.49
	$\times 0.9$	0.98	1.87	2.35	2.37
Ours	-	0.54	0.11	0.54	0.11
	$\times 0.5$	0.55	0.14	0.55	0.10
	$\times 0.7$	0.57	0.20	0.67	0.12
	$\times 0.9$	0.63	0.38	0.86	0.49

Table S.4. Zero-shot generalization across distinct driving simulators under varying degrees of ego-velocity perturbations.

Method	Velo. Noise	NAVSIM	Bench2Drive	
		PDMS ↑	DS ↑	SR (%) ↑
VAD-tiny	-	80.5	42.73	14.18
	$\times 0.0$	47.1	12.09	2.27
	$\times 0.5$	76.8	39.21	9.09
	$\times 1.5$	65.3	20.95	4.54
	$100m/s$	15.9	2.91	0.00
Ours	-	87.2	49.83	19.42
	$\times 0.0$	58.1	18.05	4.54
	$\times 0.5$	83.4	45.14	15.91
	$\times 1.5$	74.3	28.10	9.09
	$100m/s$	25.7	6.49	0.00

Table S.5. Generalization across different end-to-end learning paradigms (e.g., parallel and iterative architectures).

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
SparseDrive	0.28	0.55	0.90	0.58	0.00	0.06	0.18	0.08
+ Ours SCIS	**0.24**	**0.51**	**0.84**	**0.53**	**0.00**	**0.04**	**0.14**	**0.06**
SparseDrive-I	0.31	0.59	0.95	0.62	0.02	0.08	0.21	0.10
+ Ours SCIS	**0.27**	**0.53**	**0.86**	**0.55**	**0.01**	**0.04**	**0.16**	**0.07**

results indicate that CausalVAD maintains or slightly improves object detection (e.g., mAP) and motion prediction (e.g., minADE) compared to the VAD baseline. This confirms that our method selectively removes harmful noise (e.g., context-induced hallucinations) without discarding necessary semantic information.

4.3. Computational efficiency analysis

We prioritize the efficiency of our design to ensure it remains practical for autonomous driving. As shown in Table S.2, the integration of SCIS introduces a negligible increase in parameters (+4.7M) and inference latency (+6ms) compared to the VAD-tiny baseline. This efficiency stems from our design choice to operate on sparse, vectorized queries rather than dense feature maps, verifying that our SCIS is a lightweight and plug-and-play solution.

4.4. Baseline reproduction and analysis

To ensure a fair evaluation, we strictly follow the official VAD-tiny configuration, utilizing a ResNet-50 backbone *without* ego-status inputs. Our reproduced L2 error of 0.74m aligns with the official repository results. Notably, some literature references an L2 error of 0.41m for VAD; however, this metric corresponds to the much heavier VAD-Base model equipped with ego-status. Thus, our comparison remains rigorously fair, and the performance improvements are genuine. Regarding the collision rate, CausalVAD achieves a substantial 75% reduction compared to VAD-tiny. While methods like BridgeAD report an exceptionally low collision rate (0.08%), this is primarily attributed to its specialized collision optimization design and a heavier ResNet-101 backbone. Our framework effectively unlocks state-of-the-art safety within lightweight architectural constraints.

4.5. Validation of causal effects via counterfactual perturbation

To validate the causal effects beyond representation visualizations, we conduct interventional perturbation studies that serve as counterfactual tests. Similar to prior causal discovery benchmarks [4], we inject uniform noise of varying intensities into the Agent or Map queries to artificially amplify the spurious paths (e.g., $\mathcal{M} \to \mathcal{B} \to \mathcal{O} \to \mathcal{A} \to \mathcal{E}$ and $\mathcal{A} \to \mathcal{O} \to \mathcal{B} \to \mathcal{M} \to \mathcal{E}$). As shown in Table S.3, while the performance of VAD-tiny degrades rapidly under heavy context noise, CausalVAD maintains superior stability. This rigorous intervention quantitatively proves the effective severance of spurious links within the interaction mechanisms.

4.6. Generalization across domains and paradigms

We further demonstrate the robust generalization capabilities of our method across diverse dimensions. First, we evaluate zero-shot domain generalization. As detailed in Ta-

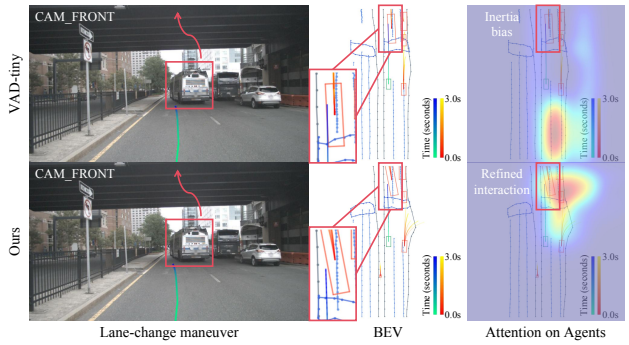

Figure S.1. Qualitative visualization of IDM in a cut-in scenario. (Top) The baseline model suffers from *inertia bias*, where the ego-query erroneously attends to the straight lane topology (spurious map context) rather than the intruding bus. (Bottom) By eliminating spurious factors from the agent features via IDM, our model disentangles the agent's motion from the map prior. This forces the attention mechanism to shift focus from the background lane to the actual cutting-in hazard, enabling safer interaction.

ble S.4, our model, trained exclusively on nuScenes, maintains commendable performance when directly deployed to the NAVSIM (non-reactive) and Bench2Drive (closed-loop) environments without any fine-tuning. This zero-shot robustness, especially under extreme ego-velocity perturbations, verifies that the model relies on universal causal mechanisms rather than dataset-specific statistics.

Second, considering that SCIS is a modular and architecture-agnostic design, we extend it to the SparseDrive framework [5] under the parallel paradigm, as well as its iterative variant (SparseDrive-I). As reported in Table S.5, integrating SCIS yields consistent performance gains across varying architectures. This confirms the broad applicability and profound potential of our causal intervention framework beyond the sequential VAD setting.

4.7. Rationale for extreme perturbation tests

In Table 3 of the main paper, we subjected the model to extreme perturbations, such as zeroing out the ego-velocity or setting it to 100 m/s. While these conditions are rare in nominal driving, they serve as a critical stress test for physical consistency. A valid causal model should exhibit increased uncertainty or planning errors when the input velocity contradicts the positional updates (a violation of physics), rather than blindly following the inertia shortcut. The robustness of CausalVAD under these conditions confirms that the model has learned to verify the ego-motion against the visual context, rather than relying solely on historical state statistics.

4.8. Qualitative visualization of de-confounding

To intuitively elucidate the operational mechanism of our proposed IDM, we analyze a representative scenario from the nuScenes validation set. This case study demonstrates how SCIS effectively identifies and severs spurious connections in complex driving environments.

IDM case study: mitigating inertia bias in cut-in scenarios. We examine a critical cut-in scenario where an agent vehicle in the adjacent lane initiates a lane change in front of the ego-vehicle. In the nuScenes dataset, vehicles traversing straight lane topologies overwhelmingly maintain a straight trajectory. This statistical regularity creates a strong spurious correlation between the Map context (straight lanes) and Agent motion (going straight), often referred to as inertia bias. In the baseline model, the Agent feature, which serves as the Key/Value for the Ego-Agent interaction, is heavily contaminated by this map prior, causing the planner to overlook lateral motion cues and predict a straight path, resulting in a collision risk. The IDM addresses this by refining the Agent feature prior to interaction. Specifically, the module identifies that the current Agent feature exhibits high affinity with the *straight-lane* prototype in the Map dictionary $\mathcal{Z}_m$. By gating and subtracting this *lane-keeping* component, the IDM effectively removes the background inertia bias. The residual, de-confounded feature vector consequently highlights subtle lateral velocity signals. As a result, the planner correctly attends to this refined representation, recognizes the cut-in intention, and generates a safe deceleration trajectory.

References

[1] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 2

[2] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 14864–14873, 2024. 2

[3] Judea Pearl. *Causality*. Cambridge university press, 2009. 1

[4] Mozhgan Pourkeshavarz, Junrui Zhang, and Amir Rasouli. Cadet: a causal disentanglement approach for robust trajectory prediction in autonomous driving. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 14874–14884, 2024. 4

[5] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE Int. Conf. Robot. Autom.*, pages 8795–8801. IEEE, 2025. 5

[6] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pages 10760–10770, 2020. 1

[7] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Int. Conf. Mach. Learn.*, pages 2048–2057. PMLR, 2015. 1