NEURAL INFORMATION PROCESSING SYSTEMS

Codebook-Guided Model Merging for Robust Test-Time Adaptation in Autonomous Driving

Huitong Yang Zhuoxiao Chen Fengyi Zhang Zi Huang Yadan Luo

UQMMLab

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Model Mcerging strategies for TTA

Test-time Adaptation in Limited Budget

- Existing test-time adaptation (TTA) methods often fail in high-variance tasks like 3D object detection due to unstable optimization and sharp minima.
- Recent model merging strategies based on linear mode connectivity (LMC) offer improved stability by interpolating between fine-tuned checkpoints, they are computationally expensive, requiring repeated checkpoint access and multiple forward passes.

We introduce CodeMerge, a lightweight and scalable model merging framework

- Prior TTA approaches typically handle shifts by aligning Batch Norm statistics, enforcing consistency through data augmentations, or minimizing sharpness via adversarial perturbations
- The core idea is to represent each finetuned checkpoint ΦΘ(t) by a compact “fingerprir derived from the source model’s penultimate c. These fingerprints serve as keys in a moc codebook, mapping to their corresponding checkpoint weights.

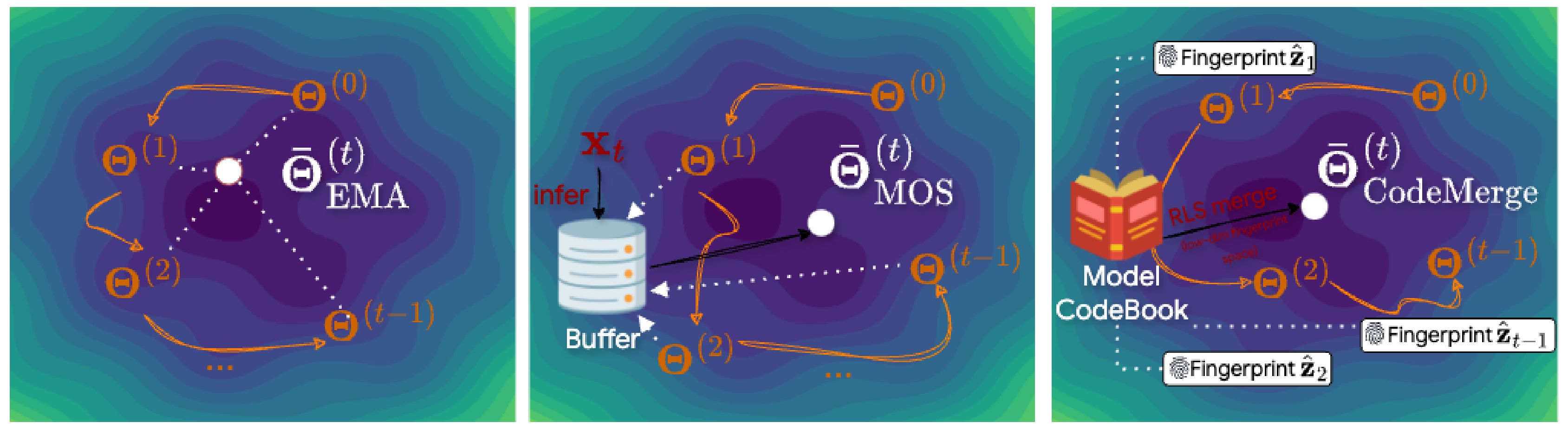

Figure 1: **Conceptual comparison of model merging strategies for TTA.** Unlike EMA (left), which ignores model behavior, or MOS (middle), which requires multiple inferences to compute merging weights, CodeMerge (right) leverages ridge leverage scores in a compact fingerprint space to efficiently guide model merging.

Model CodeBook

At each step t, we maintain a model codebook for all past checkpoints along the adaptation trajectory, denoted as:

$$\mathcal{C}^{(t)} \text{ c } \{\mathbf{z}\hat{\ } i : \Theta^{(i)}\}^{t-1}$$

Each entry is a key-value pair, where the key z ̂i ∈ Rd ′ is a low-dimensional fingerprint and the value Θ(i) is the corresponding checkpoint fine-tuned at time step i. To compute the key z ̂i, we extract intermediate features from the i-th input batch xi using a pretrained feature extractor ϕΘ(0) and randomly project them to a low-dimensional subspace for efficiency:

$$\mathbf{z}\hat{\ } i = \mathrm{RandProj}(\phi_{\Theta}(0)\ (\mathbf{x}_i)).$$

Here, RandProj(·) : Rd → Rd ′ is implemented via a fixed Gaussian projection matrix where d ′ ≪ d ensures the keys are compact. As the test-time adaptation progresses, we update the codebook incrementally by appending new pairs, i.e., C(t+1) ← (z ̂t, Θ(t)).

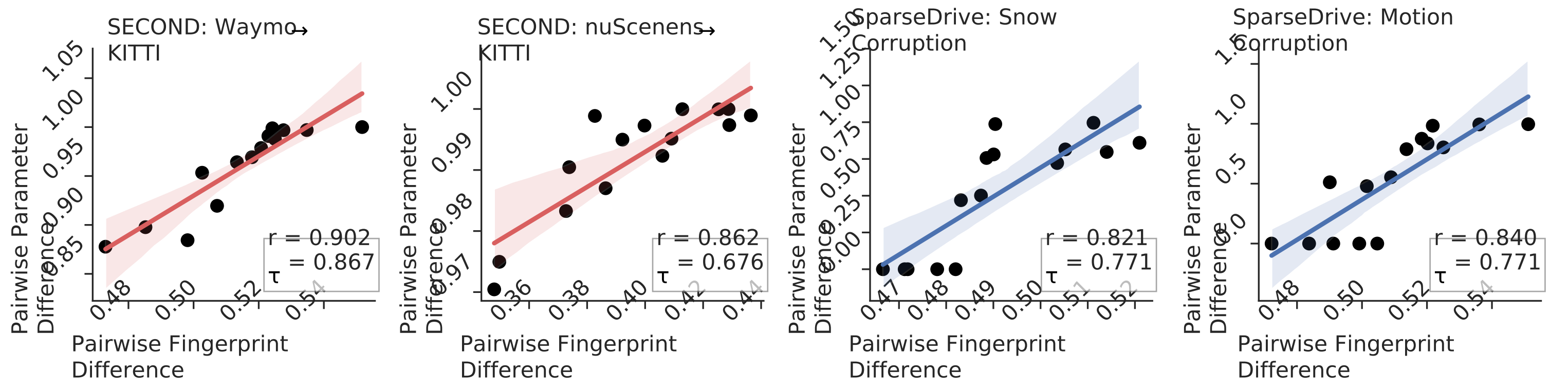

Figure 3: Pairwise fingerprint differences correlate strongly with model weight differences (Pearson r and Kendall Tau $\tau > 0.7$) across SparseDrive and SECOND , showing that the low-dimensional fingerprint space reliably reflects parameter space structure.

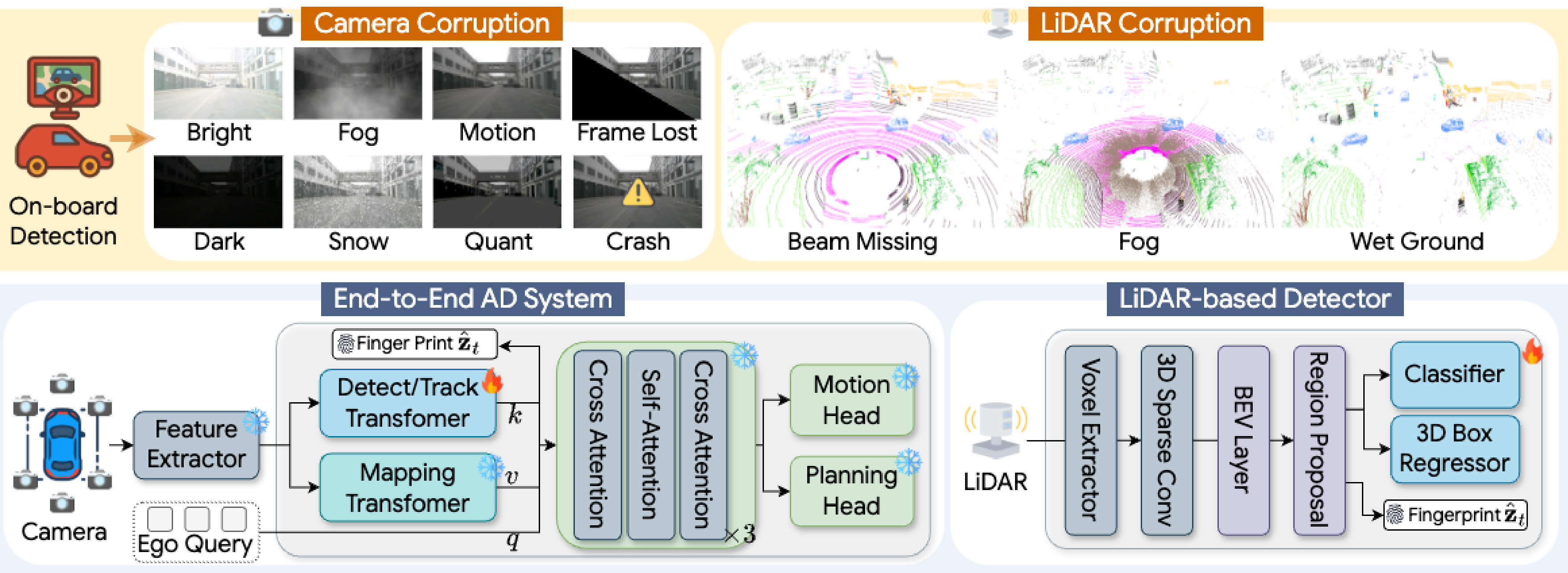

Figure 2: Overview of real-world test-time shifts (top) and 3D perception systems considered in this work (bottom). We study test-time adaptation (TTA) in two settings: (1) an end-to-end autonomous driving system and (2) a modular LiDAR-based detector, both affected by adverse weather and sensor failures. CodeMerge enables efficient TTA by leveraging compact fingerprints to guide model merging.

Curvature-Aware Merge Scores

Definition 1 (Ridge Leverage Score (RLS)). Let $\hat{\mathbf{Z}}_{t-1} = [\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_{t-1}] \in \mathbb{R}^{(t-1)\times d'}$ be the matrix of all stored keys (fingerprints), where $\hat{\mathbf{z}}_i$ be the fingerprint of the i-th candidate model $\Theta^{(i)}$. We define the ridge leverage scores of the fingerprint $\hat{\mathbf{z}}_i$ as

$$s_i^{(t)} = \hat{\mathbf{z}}_i^\top \left(\frac{1}{K}\hat{\mathbf{Z}}_{t-1}^\top \hat{\mathbf{Z}}_{t-1} + \lambda I\right)^{-1} \hat{\mathbf{z}}_i,$$

where λ is a regularization parameter. A high leverage score indicates $\hat{\mathbf{z}}_i$ is influential and lessly observed within the current feature space defined by past direction.

Theoretical Analysis. We now connect this leverage score to the inverse of curvature through the lens of LMC. We begin by revisiting the LMC assumption (Eq. (1)) through a second-order Taylor expansion around $\Theta^{(0)}$:

$$\mathcal{L}(\Theta^{(i)}) \approx \mathcal{L}(\Theta^{(0)}) + \nabla\mathcal{L}^\top \delta_i + \frac{1}{2}\delta_i{}^\top H\delta_i, \text{ with } H := \nabla^2_\theta \mathcal{L}(\Theta^{(0)}),$$

where $\delta_i := \Theta^{(i)} - \Theta^{(0)}$ refers the model update direction and $\mathbf{H}$ is the Hessian at $\Theta^{(0)}$. In this view, the curvature along δ_i is quantified by the quadratic term $\delta^\top H\delta_i$. Itcs inverse $\delta^\top H^{-1}\delta_i$ suggests δ_i explores a novel region of the loss landscape, making it an indicator for selecting diverse checkpoints. However, computing the full Hessian in high-dimensional parameter space is impractical, especially in TTA tasks. However, considering that 3D object detection models commonly use linear layers as final regression heads, we can effectively analyze curvature through the simpler and analytically tractable ridge regression setting. Specifically, assume a linear regression head parameterized by weights $w \in \mathbb{R}^d$ and a fixed feature extractor $\phi(\cdot)$, yielding a ridge regression objective of the form:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|w^\top \phi(\mathbf{x}_i) - \mathbf{y}_i\|^2 + \lambda\|w\|^2, H_w = 2(\frac{1}{K}\mathbf{Z}^\top\mathbf{Z} + \lambda I),$$

where H_w is Hessian matrix in parameter space. More precisely, this reveals the inverse of parameter-space curvature is linked to the proposed ridge leverage score under the low-rank surrogate $\hat{\ }\mathbf{Z}_1\ \hat{\ }t-1$:

$$\mathbf{z}_i^\top H_w^{-1}\mathbf{z}_i = \mathbf{z}_i^\top \left(\frac{2}{K}\mathbf{Z}_{t-1}^\top \mathbf{Z}_{t-1} + 2\lambda I\right)^{-1} \mathbf{z}_i \propto s_i^{(t)}.$$

Empirical analysis (see Fig. 3) confirms that fingerprint vectors strongly correlate (Pearson correlation and Kendall Tau scores often exceeding 0.7)with parameter deltas, confirming that the geometry of fingerprint space reliably mirrors that of parameter space.

Quantitative Results

Table 1: **Perception and tracking results** of the end-to-end SparseDrive model with and without TTA on the **nuScenes-C** validation set under different corruptions at the highest severity level. The best results for each metric and corruption are highlighted in **bold**.

Corruption	Method	3D Object Detection							Multi-Object Tracking		
		mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	AMOTA↑	AMOTP↓	Recall↑
Motion	No Adapt.	0.1468	0.3136	0.7792	0.2908	0.8048	0.4835	0.2398	0.0896	1.7983	0.1837
	Ours	**0.2759**	**0.4206**	**0.6697**	**0.2815**	0.6437	0.3618	0.2169	**0.2192**	1.5485	**0.3456**
Quant	No Adapt.	0.2022	0.3767	0.7095	0.2896	0.6478	0.3814	0.2160	0.1548	1.5398	0.2873
	Ours	**0.2742**	**0.4331**	0.6575	**0.2764**	**0.5903**	**0.3018**	**0.2137**	**0.2339**	**1.4868**	**0.3330**
Dark	No Adapt.	0.1386	0.2804	0.7375	0.4180	0.6880	0.6285	0.4164	0.1169	1.7520	0.1995
	Ours	**0.2060**	**0.3727**	**0.7206**	**0.2852**	0.6782	**0.3993**	**0.2196**	**0.1762**	**1.6333**	**0.2557**
Bright	No Adapt.	0.3300	0.4641	0.6355	**0.2749**	0.6084	0.3013	0.1892	0.2829	1.4257	0.3982
	Ours	**0.3692**	**0.4939**	0.6138	0.2779	**0.5343**	**0.2885**	0.1928	**0.3317**	**1.3389**	**0.4632**
Snow	No Adapt.	0.0970	0.2206	0.7974	0.4586	0.9349	0.6614	0.4264	0.0469	1.8822	0.1070
	Ours	**0.1828**	**0.3581**	0.7558	**0.2930**	**0.6009**	**0.4604**	**0.2222**	**0.1136**	**1.7119**	**0.2293**
Fog	No Adapt.	0.3162	0.4612	0.6295	0.2775	0.5727	0.2984	**0.1910**	0.2756	1.4469	0.3859
	Ours	**0.3421**	**0.4761**	**0.6184**	**0.2739**	0.5597	0.2995	0.1981	**0.2997**	**1.3749**	**0.4124**
Crash	No Adapt.	0.0785	0.2753	**0.6467**	0.4060	0.6078	0.5953	0.3840	0.0670	**1.8241**	0.1519
	Ours	**0.0973**	**0.3288**	0.6979	**0.2889**	**0.6061**	**0.4175**	**0.1876**	**0.0810**	1.8372	**0.1550**
Lost	No Adapt.	0.0886	0.3109	**0.7314**	0.2792	0.6206	0.4717	0.2310	0.0549	1.7638	0.1644
	Ours	**0.1172**	**0.3292**	0.7638	**0.2787**	**0.5810**	**0.4461**	**0.2243**	**0.0700**	**1.7605**	**0.1788**
Average	No Adapt.	0.1747	0.3378	0.7083	0.3368	0.6856	0.4777	0.2867	0.1361	1.6791	0.2347
	Ours	**0.2334**	**0.4016**	**0.6872**	**0.2819**	**0.5993**	**0.3719**	**0.2094**	**0.1907**	**1.5865**	**0.2966**

Table 2: **Impact of TTA on downstream modules of end-to-end SparseDrive.** We evaluate online mapping, motion prediction, and planning on the **nuScenes-C** validation set under the highest severity of various corruptions. These modules are not fine-tuned; all performance gains stem from TTA applied to the detection module. Best results per metric and corruption are shown in **bold**.

Corruption	Method	Online Mapping				Motion Prediction				Planning	
		AP_{ped}↑	AP_d↑	AP_b↑	mAP↑	mADE↓	mFDE↓	MR↓	EPA↑	L2-Avg↓	CR-Avg↓
Motion	No Adapt.	0.1988	0.2343	0.1999	0.2110	0.8630	1.3483	0.1750	0.2616	0.7877	0.2150
	Ours	**0.3660**	**0.4212**	**0.4283**	**0.4052**	**0.7264**	**1.1200**	0.1570	**0.3945**	0.6580	0.1100
Quant	No Adapt.	0.1742	0.2317	0.2069	0.2043	0.7620	1.1734	0.1526	0.3204	0.7301	0.1590
	Ours	**0.2600**	**0.3445**	**0.3267**	**0.3104**	**0.7002**	1.0859	0.1454	**0.3840**	**0.6762**	0.1250
Dark	No Adapt.	0.1173	0.2038	0.1812	0.1675	0.8428	1.3255	0.1714	0.2757	0.7535	0.2760
	Ours	**0.2825**	**0.3637**	**0.3291**	**0.3251**	**0.7493**	**1.1639**	**0.1644**	**0.3397**	**0.6602**	**0.1170**
Bright	No Adapt.	0.3777	0.4847	0.4833	0.4486	0.6646	1.0246	0.1369	0.4468	0.6306	0.1260
	Ours	**0.4305**	**0.5224**	**0.5398**	**0.4976**	0.6504	1.0122	0.1392	**0.4680**	**0.6209**	**0.0940**
Snow	No Adapt.	0.0061	0.0322	0.0369	0.0250	1.0643	1.7042	0.1930	0.2113	0.8897	0.4310
	Ours	**0.1134**	**0.1812**	**0.1740**	**0.1562**	**0.8074**	**1.2589**	**0.1717**	**0.3135**	0.7634	0.1900
Fog	No Adapt.	0.3600	0.4649	0.4076	0.4109	**0.6482**	**0.9904**	**0.1347**	0.4380	0.6257	0.1050
	Ours	**0.4276**	**0.5022**	**0.4843**	**0.4714**	0.6501	1.0008	0.1394	**0.4557**	**0.6200**	0.1100
Crash	No Adapt.	**0.1029**	0.1019	**0.0618**	**0.0889**	0.8662	1.3375	0.1652	0.1920	0.9276	**0.3740**
	Ours	0.0727	**0.1154**	0.0279	0.0720	**0.8302**	**1.3022**	**0.1637**	**0.1974**	**0.8539**	0.6300
Lost	No Adapt.	0.0892	0.0388	**0.0250**	0.0510	1.0327	1.4772	0.1740	**0.1826**	0.9932	**0.4830**
	Ours	**0.0723**	**0.0503**	**0.0250**	**0.0492**	**1.0004**	**1.4304**	**0.1739**	0.0952	**0.9600**	0.6610
Average	No Adapt.	0.1783	0.2240	0.2003	0.2009	0.8430	1.2976	0.1629	0.2911	0.7923	0.2711
	Ours	**0.2531**	**0.3126**	**0.2919**	**0.2859**	**0.7643**	**1.1718**	**0.1568**	**0.3312**	**0.7266**	0.2546

Table 3: **TTA results for LiDAR-based 3D detection across different datasets**. We report AP_{BEV} / AP_{3D} (moderate). “Oracle” = fully–supervised on target; **Bold** = best; underline = second best.

Method	Venue	TTA	Waymo → KITTI		nuScenes → KITTI	
			AP_{BEV} / AP_{3D}	Closed Gap	AP_{BEV} / AP_{3D}	Closed Gap
No Adapt.	–		67.64 / 27.48	–	51.84 / 17.92	–
SN	CVPR’20	×	78.96 / 59.20	+72.33% / +69.00%	40.03 / 21.23	+37.55% / +5.96%
ST3D	CVPR’21		82.19 / 61.83	+92.97% / +74.72%	75.94 / 54.13	+76.63% / +65.21%
Oracle	–		83.29 / 73.45	–	83.29 / 73.45	–
Tent	ICLR’21		65.09 / 30.12	–16.29% / +5.74%	46.90 / 18.83	–15.71% / +1.64%
CoTTA	CVPR’22		67.46 / 35.34	–1.15% / +17.10%	68.81 / 47.61	+53.96% / +53.47%
SAR	ICLR’23		65.81 / 30.39	–11.69% / +6.33%	61.34 / 35.74	+30.21% / +32.09%
MemCLR	WACV’23	✓	65.61 / 29.83	–12.97% / +5.11%	61.47 / 35.76	+30.62% / +32.13%
DPO	MM’24		75.81 / 55.74	+52.20% / +61.47%	73.27 / 54.38	+68.13% / +65.66%
Reg-TTA3D	ECCV’24		81.60 / 56.03	+89.20% / +62.11%	68.73 / 44.56	+53.70% / +47.97%
MOS	ICLR’25		81.90 / 64.16	+91.12% / +79.79%	71.13 / 51.11	+61.33% / +59.78%
Ours	**–**		**84.62 / 66.31**	**+108.50% / +84.47%**	**77.41 / 58.54**	**+81.30% / +73.15%**

Github: https://github.com/UQHTy/CodeMerge

NeurIPS 2025 Poster Session