WiTTA-Bench: Benchmarking Test-Time Adaptation for WiFi Sensing

Supplementary Material

Contents of Appendix

A Extended Related Work 1
A.1 WiFi Sensing 1
A.2 Domain Adaptation and Generalization 2
A.3 Test-Time Adaptation 2
A.4 Benchmarking and Evaluation Protocols 3

B Datasets 3
B.1 WiHAR-Dual Dataset 3
B.2 CSLOS Dataset 5

C Implementation Details 6

D Extended Benchmark Results 12
D.1 Feature-Space Visualization under Domain Shifts 12
D.2 Class-Level Feature Inconsistency at Test Time . 13
D.3 Measure Domain Discrepancy via MMD 13
D.4 WiTTA-Benchmarking Result Tables 13

E Extended Experiments on Factors Influencing TTA Effectiveness 14
E.1 Hyperparameter Sensitivity 14
E.2 Continual Adaptation Analysis 14
E.3 Backbone Generality Analysis 15

A. Extended Related Work

A.1. WiFi Sensing

WiFi sensing has emerged as a promising paradigm for passive sensing by leveraging the ubiquitous WiFi infrastructure. Unlike camera- or wearable-based approaches that require dedicated sensors or active user participation, WiFi sensing exploits existing wireless signals, enabling low-cost, non-intrusive, and privacy-preserving sensing. WiFi sensing leverages the RSSI or Channel State Information (CSI) embedded in WiFi signals to capture multipath propagation variations caused by human motion and environmental dynamics. Such information-rich signals encode spatial, temporal, and frequency characteristics of the surrounding environment, enabling a wide range of downstream tasks including HAR, gesture recognition, localization, and health monitoring [27].

A.1.1. Early Statistical and Signal-Processing Methods

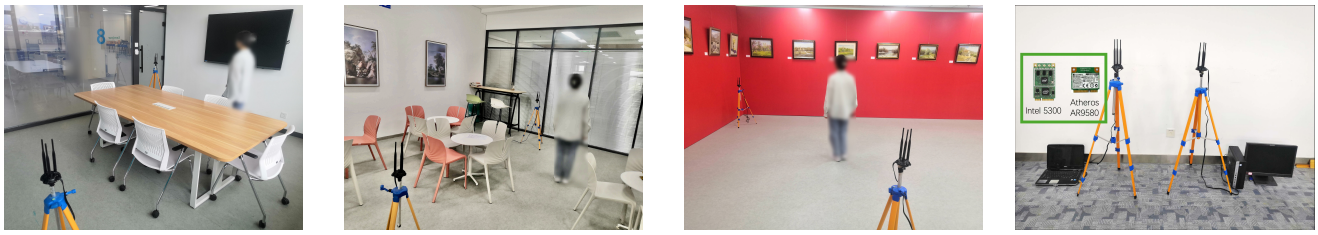
Early WiFi sensing studies primarily relied on signal processing and handcrafted feature extraction to obtain discriminative information from CSI or RSSI signals. WiFi-ID [56] utilized walking-induced temporal patterns of CSI amplitudes for human identification, while Magicol [39] fused WiFi and magnetic-field anomalies to achieve infrastructure-light indoor localization. Later, frameworks like CrossSense [57] and Widar3.0 [58] improved scalability and cross-environment generalization by modeling Doppler and multipath signatures. These methods relied on manually engineered features, such as subcarrier variance, amplitude dynamics, or CSI correlations, and traditional classifiers (e.g., SVM, KNN, Random Forest). Although effective in controlled scenarios, their performance often degraded under environmental changes or multi-person interference.

A.1.2. Deep Learning Based Approaches

The advance of deep learning has transformed WiFi sensing into a representation-learning paradigm, where models automatically learn discriminative features from raw or preprocessed CSI. DeepFi [48] demonstrated one of the first deep architectures for indoor localization using CSI amplitudes, while recent works employ convolutional and attention-based models to capture spatio-temporal dependencies. Li *et al.* proposed THAT [18] to jointly model temporal and spatial correlations across subcarriers, achieving robust activity recognition. Building on this, Difformer [19] introduced multi-resolution differencing for adaptive temporal encoding, and GrapHAR [29] leveraged graph-based subcarrier modeling for lightweight real-time inference. At the system level, SenseFi [53] established a large-scale benchmark for deep WiFi sensing. Despite these advances, deep WiFi models remain highly sensitive to domain variations, often overfitting to environment-specific propagation patterns [18, 57].

A.1.3. Domain Shift in WiFi CSI

A persistent bottleneck in WiFi sensing is its vulnerability to environmental and hardware variations, collectively known as the *domain shift* problem. The fine-grained multipath patterns of CSI are extremely sensitive to environmental geometry, device configuration, and human presence. Even subtle changes in layout, antenna orientation, or transceiver placement can distort CSI distributions, leading to substantial performance degradation in unseen domains [46, 57]. Ma *et al.* [5] reveals that CSI features are inherently non-stationary and heavily affected by multipath dynamics, underscoring the need for adaptive learning mechanisms. Wang *et al.* [46] further revealed the "WiFi sensing dilemma," showing that models collapse under domain variation and proposed conformal prediction to quantify uncertainty. Meanwhile, Wi-AM [50], AdaWiFi [63],



(a) Seminar Room (E1) (b) Lounge Space (E2) (c) Exhibition Space (E3) (d) Platform

Figure 9. Illustrations of the three data collection environments and the WiFi sensing platform used in the WiHAR-Dual dataset.

and FewSense [55] addressed cross-domain generalization for gesture and activity recognition via domain-aware and few-shot adaptation strategies. However, most existing solutions require access to both domain data during training, which is infeasible in online or privacy-constrained deployments.

A.2. Domain Adaptation and Generalization

Domain adaptation (DA) and domain generalization (DG) enhance model robustness under distributional shifts between training (source) and deployment (target) domains. DA focuses on adapting models to a specific unseen domain using limited or unlabeled target data, while DG aims to learn domain-invariant representations that generalize across multiple unseen domains.

A.2.1. Domain Adaptation

DA assumes access to both domains' data and seeks to minimize the domain gap through explicit alignment strategies. Discrepancy-based methods, such as Deep Adaptation Network (DAN) [25] and Joint Adaptation Network (JAN) [26], reduce statistical divergence (e.g., MMD) between source and target features. Adversarial approaches like DANN [10] and ADDA [43] employ gradient reversal or domain discriminators to learn domain-invariant embeddings. Motiian *et al.* [31] unified supervised DA and DG with pairwise alignment losses, and Mancini *et al.* [28] further introduced AdaGraph, leveraging domain metadata for continuous adaptation.

A.2.2. Domain Generalization

Unlike DA, DG assumes no access to target-domain data during training and focuses on learning domain-invariant representations that generalize to unseen environments. Classical DG approaches can be broadly grouped into three categories [45, 64]. *Data-centric* methods enrich source diversity to simulate unseen shifts, e.g., MixStyle [65] and cross-gradient augmentation [38]. *Representation-centric* approaches seek invariant features via disentanglement or meta-learning, such as MetaReg [2] and domain-specific normalization [8]. *Optimization-centric* schemes improve robustness through episodic or contrastive regularization [17, 20]. These studies highlight that, even without target data, implicit alignment through diverse training and structured objectives can effectively mitigate domain shifts.

There are two differences between TTA and DA/DG. First, TTA makes no strong assumptions about the nature or structure of distribution shifts, unlike Domain Generalization methods that require explicit modeling of inter-domain relationships [12]. Second, TTA operates without requiring simultaneous access to training and test data, a fundamental constraint in domain adaptation methods [10]. At its core, TTA defines an auxiliary objective at inference to adapt the model in an unsupervised manner, enabling dynamic, on-the-fly adjustment to unseen data distributions.

A.3. Test-Time Adaptation

TTA aims to enhance model robustness under distribution shifts by enabling *on-the-fly* model updates after deployment without accessing source data. Existing methods can be broadly categorized into *Online Test-Time Adaptation (OTTA)* and *Test-Time Domain Adaptation (TTDA)* depending on whether the model adapts continuously on streaming data or performs one-shot adaptation on a fixed test batch.

Early OTTA methods such as BN Calibration [30, 37, 60] updated the statistics or affine parameters of Batch Normalization layers to align feature distributions, providing computational efficiency but limited adaptation capacity. Entropy-based optimization was later introduced by TENT [44], SAR [33], which minimizes prediction entropy to achieve self-tuning adaptation. Subsequent pseudo-labeling approaches, including T3A [15] and TAST [16], refined decision boundaries through self-training guided by confidence. To ensure stable adaptation under continuously shifting test distributions, CoTTA [47] and EATA [32] mitigate catastrophic forgetting through stochastic parameter restoration and Fisher-weighted importance regularization, respectively, while RMT [7] and PETAL [3] enforce consistency regularization via teacher-student frameworks with contrastive learning and augmentation-averaged pseudo-labeling. These OTTA methods collectively address the challenge of reliable, streaming adaptation without batch access or source data.

In contrast, TTDA methods assume access to an unlabeled test set for batch-level adaptation. Representative pseudo-labeling methods such as SHOT [22], ASL [52],

and BMD [35] fine-tune classifier heads using self-generated labels. Consistency regularization methods such as APA [41] and SFDA-UR [34] enforce prediction stability through adversarial perturbations and uncertainty reduction, while clustering-based approaches such as ASFA [49] and ISFDA [21] refine decision boundaries via teacher-student distillation and prototype-based iterative refinement to address class imbalance. Meanwhile, BAIT and DaC [54, 59] estimate source knowledge through frozen classifiers and feature memory banks, respectively, to anchor target adaptation. Recent studies such as SHOT++ [23] and RULER [62] also integrate self-supervised learning to improve characteristic discriminability and perform hypothesis transfer through information maximization.

It is worth noting that, source-prepared TTA methods such as TTT [42], TTT+ [24], and recent WiFi-specific DATTA [40], relying on modifying the *source-domain* training procedure (e.g., via auxiliary self-supervised objectives) to gain adaptability. Such source-prepared designs violate the *strict TTA assumption* and are impractical in WiFi sensing, where source data are unavailable. For fairness, we exclude these methods from comparison.

In conclusion, TTA unifies the objectives of domain adaptation and generalization by performing unsupervised, inference-time optimization to handle unseen or continuously shifting domains. Its independence from source data and flexibility under dynamic test conditions make it particularly appealing for real-world, privacy-sensitive applications such as WiFi sensing.

A.4. Benchmarking and Evaluation Protocols

TTA holds particular significance for real-world WiFi HAR systems, where distribution shifts naturally arise from changes in environment layouts, device hardware, or user behavior. Unlike traditional offline retraining or domain adaptation, TTA enables on-the-fly model calibration during deployment, allowing HAR systems to maintain robustness under dynamic and privacy-sensitive conditions without requiring source data. This makes TTA particularly valuable for ubiquitous sensing applications such as smart homes, healthcare monitoring, and industrial IoT, where models must continuously adapt to non-stationary wireless channels and unseen users.

Despite this potential, research on TTA for WiFi sensing remains scarce. To the best of our knowledge, there exists *no standardized benchmark* dedicated to evaluating TTA methods in WiFi HAR. Existing efforts such as SenseFi [53] and CrossSense [57] focus on performance itself or cross-domain generalization within WiFi sensing, but do not benchmarking TTA. Albeit there are several vision TTA benchmarks, such as UniTTA [9], BoTTA [6], and TTAB [61], they primarily target *visual* (and typically, synthetic) distribution shifts (e.g., style, crop, or color perturbations) rather than the complex, physics-driven domain variations in WiFi signals that stem from uncontrolled propagation, multipath, and hardware diversity (see §3.2). Therefore, a dedicated WiFi TTA benchmark is urgently needed to systematically study adaptation behavior under realistic and uncertain domain shifts.

To bridge this gap, WiTTA-Bench extends these efforts by introducing a benchmark for wireless sensing, where domain shifts originate from multipath propagation rather than appearance variation. WiTTA-Bench defines three *propagation-induced* shifts (CE, CS, CD) under standardized OTTA and TTDA protocols to enable fair, reproducible, and extensible assessment of TTA algorithms for wireless sensing tasks.

B. Datasets

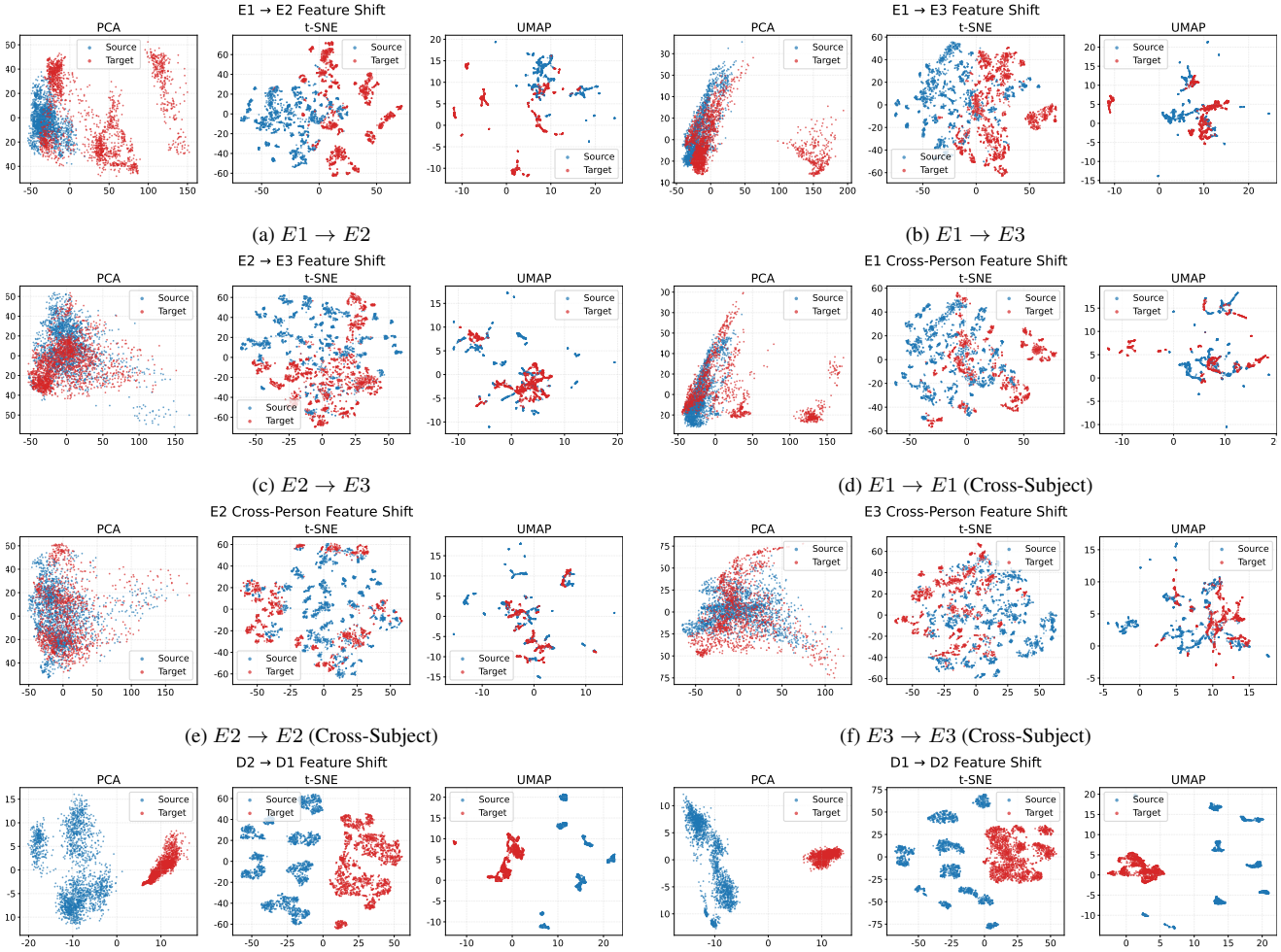
B.1. WiHAR-Dual Dataset

While existing public WiFi CSI datasets mainly cover *CE* and *CS* settings, to the best of our knowledge, none provides standardized *CD* evaluation. However, device heterogeneity poses a fundamental challenge in real-world WiFi sensing: user-end WiFi devices (NICs) often differ from the source-domain hardware, creating device-level mismatches that distort CSI distributions and hinder model transferability. To address this overlooked yet critical gap, we construct WiHAR-Dual, a large-scale dataset comprising 14,847 CSI records collected using two heterogeneous NIC platforms (Intel 5300 and Atheros AR9580) across three environments and seven daily activities. This dataset establishes, for the first time, a reproducible *CD* benchmark for WiTTA task, enabling systematic analysis of model robustness under device-induced domain shifts.

B.1.1. Devices

To assess cross-device generalization under hardware heterogeneity, we collected data using two WiFi platforms (NICs), i.e., Intel 5300 and Atheros AR9580, under identical environmental and activity setups. Both operated at 5 GHz with 20 MHz bandwidth (IEEE 802.11n) and used tripod-mounted antennas at 1.2 m height. The TX and RX were placed 4 m apart to ensure stable line-of-sight transmission. The platform is illustrated in Fig. 9 (d).

- **Intel 5300 (D1)** The Intel 5300 NICs served as the primary platform in WiHAR-Dual. CSI was extracted via `Linux 802.11n CSI Tool` [13] on Ubuntu 14.04, capturing 30 subcarriers per antenna at a sampling rate of 500 Hz. Each 4-second recording produced a CSI segment of size 2000×30 (averaging three antenna). This configuration has been widely used in previous WiFi sensing datasets and offers a unified, reproducible setting for a fair comparison.
- **Atheros AR9580 (D2)** For cross-device benchmarking, we collect parallel data using Atheros AR9580 mod-



(a) $E1 \rightarrow E2$ (b) $E1 \rightarrow E3$

(c) $E2 \rightarrow E3$ (d) $E1 \rightarrow E1$ (Cross-Subject)

(e) $E2 \rightarrow E2$ (Cross-Subject) (f) $E3 \rightarrow E3$ (Cross-Subject)

(g) Atheros → Intel (Cross-Device) (h) Intel → Atheros (Cross-Device)

Figure 10. Extended **feature-space visualizations** under CE, CS, and CD shifts. Each subfigure shows PCA, t-SNE, and UMAP embeddings of source (blue) and target (red) features, revealing varying degrees of overlap and manifold deformation across shift types.

ules configured identically to Intel 5300 (D1). The receiver employed a modified `Atheros-CSI-Tool` [51] on Ubuntu 18.04 to extract CSI across 56 subcarriers per antenna pair at a sampling rate of 500Hz, and 4-second segment thus forms a 2000×56 matrix.

B.1.2. Subjects and Activities

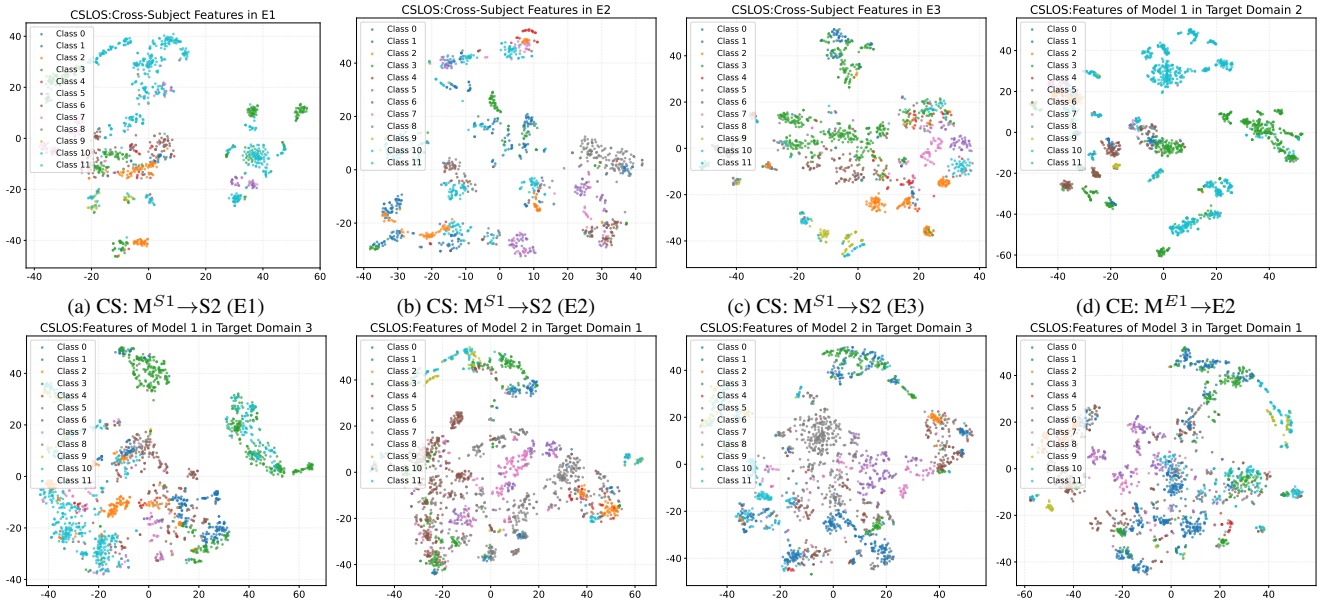
Seven volunteers (four males and three females, aged 18–27) participated. Each subject performed seven activities: *standing still*, *walking*, *running*, *bending*, *sitting down*, *jumping*, and *waving hand*. Each activity was repeated 101 times per environment, yielding 4,949 labeled samples per domain and 14,847 samples in total. The participants were instructed to move naturally to preserve realistic intra-class variance and motion dynamics.

B.1.3. Environmental Setup

Data collection was carried out in three distinct indoor environments to emulate realistic domain shifts:

- **Seminar Room (E1)** – Compact 4×3 m room with tiled floor and wooden tables, producing dense multipath reflections in a confined space (Fig. 9 (a)).
- **Lounge Space (E2)** – Medium 10×8 m area with wooden floors and mixed furniture, creating moderate reflection and human scattering Fig. 9 (b).
- **Exhibition Space (E3)** – Large 18.5×5.5 m corridor-like area with concrete floor and glass walls, dominated by LOS propagation and sparse clutter (Fig. 9 (c)).

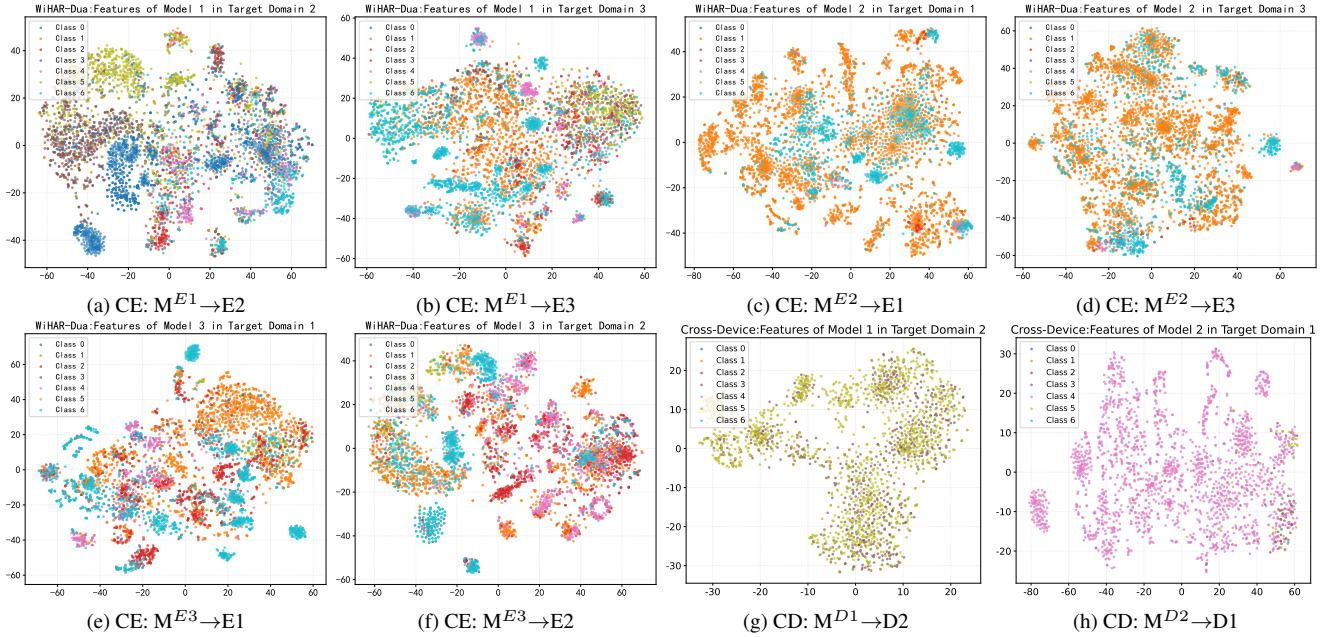
Dataset Design Goals. WiHAR-Dual provides the *first large-scale cross-device dataset* for WiFi TTA. It is designed to: *i*) evaluate physics-driven domain shifts, especially the device-induced shift; *ii*) enable controlled and reproducible evaluation in all CE, CS, and CD settings; and



(a) CS: M^{S1}→S2 (E1) (b) CS: M^{S1}→S2 (E2) (c) CS: M^{S1}→S2 (E3) (d) CE: M^{E1}→E2

(e) CE: M^{E1}→E3 (f) CE: M^{E2}→E1 (g) CE: M^{E2}→E3 (h) CE: M^{E3}→E1

Figure 11. t-SNE visualization of source-model features under CS and CE settings on **CSLOS**. Each color denotes a predicted activity class on the target domain. M^{E1}–M^{E3} correspond to models pretrained on the three environments; M^{S1}-M^{S2} indicates cross-subject transfer.



(a) CE: M^{E1}→E2 (b) CE: M^{E1}→E3 (c) CE: M^{E2}→E1 (d) CE: M^{E2}→E3

(e) CE: M^{E3}→E1 (f) CE: M^{E3}→E2 (g) CD: M^{D1}→D2 (h) CD: M^{D2}→D1

Figure 12. t-SNE visualization of source-model features under CE and CD settings on **WiHAR-Dual**. Each color denotes a predicted activity class on the target domain. M^{E1}–M^{E3} correspond to models pretrained on three environments, while M^{D1}–M^{D2} represent Intel and Atheros devices. Compared with CE, CD induces substantially larger manifold deformation due to hardware-induced distortions.

iii) support fair standardized comparison across adaptation protocols.

B.2. CSLOS Dataset

The CSLOS dataset [1] provides a large-scale benchmark for WiFi–based HAR under both line-of-sight (LOS) and non-line-of-sight (NLOS) indoor conditions. It contains 3,000 trials collected from 30 subjects (28 male, 2 female) performing 12 representative activities, including *walking*, *sitting/standing*, *falling*, *turning*, and *picking up*, among others, across three distinct indoor environments, including Lab Room (LOS), Corridor (LOS), Wooden-Partition Room

Table 7. **WiHAR-Dual OTTA Benchmark (Cross-Environment Setting).** Comparison of *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Environment (CE)** protocol. We report classification accuracy (%), FLOPs, trainable parameters, and inference latency. **PETAL** achieves the best overall trade-off between accuracy and efficiency.

Metric	Source	Target	Base	BN Calib.		Entropy Min.		Pseudo-Labeling		Anti-Forget. Reg.		Consistency Reg.	
			Base	DUA	DELTA	TENT	SAR	T3A	TAST	CoTTA	EATA	RMT	PETAL
A. Effectiveness (Accuracy %)													
Accuracy	E1	E1	99.43	99.47	99.52	99.52	99.23	98.12	92.73	99.43	99.43	99.23	**99.54**
	E2	E1	20.31	21.72	21.88	22.57	14.93	**27.18**	23.66	18.29	23.10	20.17	22.61
	E3	E1	38.63	37.93	37.04	37.70	**47.00**	40.11	33.16	15.36	38.17	37.72	38.41
	E1	E2	42.88	52.90	52.13	52.15	19.80	**54.29**	48.47	51.71	52.13	52.82	52.37
	E2	E2	99.31	99.39	99.27	99.27	**99.41**	96.81	93.27	99.37	99.33	98.99	99.11
	E3	E2	**35.10**	31.78	31.04	31.60	33.22	31.26	22.81	15.96	31.46	33.08	31.68
	E1	E3	24.09	42.21	42.88	43.60	**44.84**	44.62	40.49	41.16	43.91	42.74	44.25
	E2	E3	24.03	25.88	26.15	26.37	18.55	**29.76**	26.13	14.51	26.29	24.97	26.53
	E3	E3	99.09	99.09	98.93	98.89	98.38	97.29	81.25	**99.11**	98.91	98.32	98.89
B. Efficiency													
FLOPs (M)	All	All	559.94	1754.47	552.90	559.94	552.90	559.94	559.94	19597.82	559.94	2231.56	19597.82
Params	All	All	0	0	480	480	480	0	16768	4.37M	480	12.69M	4.37M
Latency (ms)	All	All	0.092	0.149	0.393	0.232	0.745	0.203	0.358	70.686	0.233	1.146	3.437

Table 8. **WiHAR-Dual TTDA Benchmark (Cross-Environment Setting).** Comparison of *Test-Time Domain Adaptation* (TTDA) methods under the **Cross-Environment (CE)** protocol. We report classification accuracy (%), FLOPs, trainable parameters, adaptation time, and inference latency.

Metric	Source	Target	Base	Pseudo-Labeling			Consistency		Clustering		Source Est.		Self-Supervision
			Base	SHOT	ASL	BMD	APA	SFDA-UR	ASFA	ISFDA	BAIT	DaC	SHOT++
A. Effectiveness (Accuracy %)													
Accuracy	E1	E1	99.43	99.45	99.52	99.13	99.37	99.54	**99.56**	99.43	99.21	99.52	99.31
	E2	E1	20.31	36.26	35.99	60.48	28.57	23.92	61.71	50.35	**68.68**	24.23	63.55
	E3	E1	38.63	68.42	56.66	69.32	36.03	40.04	73.69	51.10	64.32	45.52	**86.58**
	E1	E2	42.88	77.77	55.93	58.82	48.21	53.97	74.54	64.68	59.67	54.29	**91.37**
	E2	E2	99.31	99.74	99.52	97.69	99.68	99.45	**99.76**	99.37	98.36	99.37	99.74
	E3	E2	35.10	**72.12**	38.55	54.29	38.78	36.94	68.05	44.76	65.99	32.37	71.33
	E1	E3	24.09	41.16	48.41	57.41	40.98	44.37	44.76	26.39	50.60	47.71	**71.91**
	E2	E3	24.03	33.93	35.79	53.59	40.15	28.27	35.72	37.83	38.13	29.82	**63.31**
	E3	E3	99.09	99.13	99.15	98.69	99.03	99.11	**99.17**	**99.17**	99.01	99.09	98.89
B. Efficiency Metrics													
FLOPs (M)	All	All	559.94	559.94	559.94	559.94	552.90	559.94	559.94	552.90	559.94	559.94	559.94
Params (M)	All	All	0	0.27	4.37	4.10	4.37	4.38	4.37	4.37	4.37	4.37	4.37
Adaptation Time (s)	All	All	0	153.35	289.55	45.82	402.05	129.60	155.52	130.99	390.90	296.78	775.17

(NLOS). WiFi CSI was recorded using Intel 5300 NICs operating at 2.4 GHz, 20 MHz bandwidth, and 320 packets/s sampling rate. The setup forms a 1×3 MIMO system (one TX, three RX antennas), yielding 30×3 CSI subcarriers per packet. The first two environments adopt LOS configurations (a 4.7 m × 4.7 m lab and a 7.95 m × 3.6 m corridor), whereas the third introduces a wooden wall barrier (8 cm thick, 5.44 m TX–RX distance) to emulate NLOS propagation.

This configuration enables systematic evaluation across diverse propagation conditions and makes CSLOS a challenging benchmark for WiFi sensing robustness under environmental shifts.

C. Implementation Details

Computing Platform. All methods adopt their official implementations and are standardized to a unified Python and PyTorch versions (PyTorch 2.7.1 with Python 3.9.23) for consistency across experiments. The experiments were conducted on a server equipped with an Intel Xeon Gold 6530 CPU, $8 \times$ NVIDIA GeForce RTX 5090 GPUs with 32GB VRAM each, 1TB RAM, and running on Ubuntu 22.04 LTS.

Data Preprocessing. For the WiHAR-Dual dataset, each record collected using Intel 5300 NIC has dimensionality of 2000×30. The Atheros NIC captures 56 subcarri-

Table 9. **CSLOS OTTA Benchmark (Cross-Environment Setting).** Comparison of *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Environment (CE)** protocol on the CSLOS dataset. We report classification accuracy (%), FLOPs, trainable parameters, and inference latency.

Metric	Source	Target	Base	BN Calib.		Entropy Min.		Pseudo-Labeling		Anti-Forget. Reg.		Consistency Reg.	
			Base	DUA	DELTA	TENT	SAR	T3A	TAST	CoTTA	EATA	RMT	PETAL
A. Effectiveness (Accuracy %)													
Accuracy	E1	E1	98.37	**98.37**	97.53	97.77	97.74	97.83	91.72	97.65	97.29	97.62	98.22
	E2	E1	29.26	30.25	29.56	28.84	29.44	**32.06**	27.87	25.17	29.38	29.89	31.16
	E3	E1	41.15	41.93	40.88	40.94	42.29	40.55	40.07	35.28	41.66	41.84	**42.47**
	E1	E2	27.18	**29.89**	28.60	28.88	29.06	23.61	27.77	22.62	28.54	29.34	**29.89**
	E2	E2	98.18	**98.24**	97.90	97.63	97.53	95.22	88.81	97.60	97.53	97.72	97.44
	E3	E2	26.22	30.05	28.44	28.94	28.75	28.60	29.55	17.57	29.65	29.58	**30.35**
	E1	E3	38.06	39.43	36.72	37.56	38.36	36.87	33.14	29.84	36.75	38.00	**39.55**
	E2	E3	30.67	31.48	30.97	31.06	31.30	**33.98**	31.06	18.96	31.24	30.49	32.34
	E3	E3	**98.81**	98.78	98.51	98.60	98.48	94.90	89.39	98.39	98.33	98.45	98.72
B. Efficiency Metrics													
FLOPs (M)	All	All	286.69	573.38	283.08	286.69	286.69	286.69	286.69	9747.47	286.69	1433.45	9747.47
Params	All	All	0	0	480	480	480	0	17920	2370000	480	6820000	2370000
Latency (ms)	All	All	0.37	0.38	1.44	0.39	0.47	0.38	0.42	186.21	0.39	0.43	39.52

Table 10. **CSLOS TTDA Benchmark (Cross-Environment Setting).** Comparison of *Test-Time Domain Adaptation* (TTDA) methods under the **Cross-Environment (CE)** protocol on the CSLOS dataset. We report classification accuracy (%), FLOPs, trainable parameters, adaptation time, and inference latency.

Metric	Source	Target	Base	Pseudo-Labeling			Consistency		Clustering		Source Est.		Self-Supervision
			Base	SHOT	ASL	BMD	APA	SFDA-UR	ASFA	ISFDA	BAIT	DaC	SHOT++
A. Effectiveness (Accuracy %)													
Accuracy	E1	E1	98.37	98.59	98.07	95.99	98.37	99.13	98.89	98.52	**99.71**	98.65	98.98
	E2	E1	29.26	32.96	32.84	31.23	32.36	33.11	32.93	33.53	**34.80**	32.06	33.77
	E3	E1	41.15	41.78	44.97	41.16	44.82	41.45	42.26	38.38	41.73	**46.33**	45.48
	E1	E2	27.18	31.65	31.56	30.82	29.21	30.60	31.03	25.49	**37.44**	30.32	32.08
	E2	E2	98.18	98.21	97.75	96.58	98.31	**99.04**	98.86	98.40	99.02	97.97	98.67
	E3	E2	26.22	27.33	32.57	27.85	30.82	29.46	32.33	26.35	**33.90**	31.99	32.23
	E1	E3	38.06	38.99	**41.31**	40.04	41.01	39.34	41.28	38.63	41.22	40.36	40.89
	E2	E3	30.67	33.29	34.31	34.01	**35.89**	33.38	34.40	34.72	34.47	33.50	35.05
	E3	E3	98.81	98.84	98.57	97.92	98.81	98.99	98.90	98.84	**99.32**	98.57	98.60
B. Efficiency Metrics													
FLOPs (M)	All	All	286.69	288.55	1433.76	286.69	286.69	286.69	286.69	283.09	286.69	286.69	286.69
Params (M)	All	All	0	0.27	2.37	2.10	2.37	2.38	0.27	2.37	2.37	2.37	2.37
Adaptation Time (s)	All	All	0	149.07	69.63	41.09	56.38	112.47	38.51	36.23	97.02	122.38	233.89

ers, and we apply padding and average pooling to reduce them to 30 subcarriers, aligning the dimensionality with that of the Intel 5300 data and resulting in samples of size 2000×30. The CSLOS dataset contains three data streams with 30 subcarriers each; we use only the first stream in our experiments, but its temporal length varies across samples. To ensure consistency, we apply resampling and truncation to standardize all samples to 1024×30.

Model. All methods employ an identical backbone to ensure strict fairness and reproducibility. The model consists of a CNN encoder followed by a two-layer MLP classifier. The encoder comprises four sequential convolutional blocks, each containing a Conv2d layer (kernel size 5×5, stride 1, padding 2, with $[16, 32, 64, 128]$ output channels across the four blocks), a BatchNorm2d, a ReLU activation, and a MaxPool2d layer (kernel size 2, stride 2) for spatial downsampling. After four blocks, the encoder produces a feature map of size $(128, \lfloor H/16 \rfloor, \lfloor W/16 \rfloor)$ for an input of shape $(1, H, W)$. The feature map is flattened and passed through a two-layer MLP classifier having 256 hidden units with ReLU and dropout (rate 0.5), followed by a final linear layer that outputs the class logits.

Optimization. In the cross-environment setting, base models were trained per environment using Adam (learning rate 1×10^{-4}, weight decay 1×10^{-5}, batch size 30) for 100 epochs with an 8:2 train–test split, selecting the check-

Table 11. **CSLOS OTTA Benchmark (Cross-Subject Setting).** Comparison of *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Subject (CS)** protocol on the CSLOS dataset. We report classification accuracy (%), FLOPs, trainable parameters, and inference latency.

Metric	Cross-Subject	Base	BN Calib.		Entropy Min.		Pseudo-Labeling		Anti-Forget. Reg.		Consistency Reg.	
		Base	DUA	DELTA	TENT	SAR	T3A	TAST	CoTTA	EATA	RMT	PETAL
A. Effectiveness (Accuracy %)												
Accuracy	E1	97.69	**97.69**	97.54	97.44	97.19	93.98	91.12	97.19	97.24	96.89	96.44
	E1 cross-subject	33.71	32.66	33.78	33.71	34.01	31.75	27.46	19.26	**35.06**	33.94	32.96
	E2	97.30	97.25	97.50	97.35	96.99	97.30	95.16	96.99	**97.60**	96.79	96.94
	E2 cross-subject	38.47	39.02	38.08	39.10	39.64	34.58	40.03	36.60	**40.11**	38.32	38.63
	E3	98.20	98.05	**98.35**	98.10	97.96	95.16	95.06	97.96	**98.35**	97.41	97.46
	E3 cross-subject	33.56	33.11	33.48	33.63	33.78	31.04	33.04	32.96	33.48	34.00	**35.33**
B. Efficiency Metrics												
FLOPs (M)	All	286.69	573.38	283.08	286.69	286.69	286.69	286.69	9747.47	286.69	1433.45	9747.47
Params	All	0	0	480	480	480	0	17920	2370000	480	6820000	2370000
Latency (ms)	All	0.39	0.97	9.58	2.23	0.63	6.74	36.60	249.20	7.35	17.28	36.91

Table 12. **CSLOS TTDA Benchmark (Cross-Subject Setting).** Comparison of *Test-Time Domain Adaptation* (TTDA) methods under the **Cross-Subject (CS)** protocol on the CSLOS dataset. We report classification accuracy (%), FLOPs, trainable parameters, adaptation time, and inference latency.

Metric	Cross-Subject	Base	Pseudo-Labeling			Consistency		Clustering		Source Est.		Self-Supervision
		Base	SHOT	ASL	BMD	APA	SFDA-UR	ASFA	ISFDA	BAIT	DaC	SHOT++
A. Effectiveness (Accuracy %)												
Accuracy	E1	97.69	**98.44**	98.04	95.28	97.69	97.79	97.79	97.89	98.08	97.99	98.09
	E1 cross-subject	33.71	**39.28**	34.91	31.67	38.37	34.16	33.86	37.25	32.40	34.39	35.14
	E2	97.30	98.37	96.84	96.50	97.30	98.32	98.11	97.81	**98.65**	97.20	98.27
	E2 cross-subject	38.47	41.20	39.56	41.05	41.12	39.72	41.90	39.25	43.71	39.88	**47.43**
	E3	98.20	98.45	97.61	97.13	98.20	98.55	98.40	98.50	**98.85**	98.10	98.30
	E3 cross-subject	33.56	33.63	34.37	**39.28**	35.41	33.78	32.52	32.74	36.88	33.63	**39.28**
B. Efficiency Metrics												
FLOPs (M)	All	286.69	288.55	1433.76	286.69	286.69	286.69	286.69	283.09	286.69	286.69	286.69
Params (M)	All	0	0.27	2.37	2.10	2.37	2.38	0.27	2.37	2.37	2.37	2.37
Adaptation Time (s)	All	0	80.82	68.16	21.74	113.64	43.93	23.75	20.96	49.75	76.73	83.08

point with the best validation accuracy. In the cross-subject case, six subjects act as the source and four as the target on the CSLOS dataset (learning rate 5×10^{-5}, weight decay 1×10^{-4}, 50 epochs). In the cross-device case, we use one device and adapting to the other (WiHAR-Dual) under the same configuration as the cross-environment setup.

Adaptation Details of OTTA. Hyperparameters remain fixed per setting during adaptation. All OTTA methods perform a single forward–backward–update per incoming batch (batch size 30) without replay, except TAST (steps $= 2$). Crucially, these methods maintain persistent parameter updates throughout the target domain traversal; parameters are never reset except under the *episodic* setting, where the model is re-initialized for every incoming batch.

Most OTTA baselines are BN-centric and update only the BatchNorm affine parameters. We optimize them using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0) with method-specific learning rates. Entropy-minimization methods include TENT (learning rate 5×10^{-4}) and EATA which extends TENT by filtering unreliable high-entropy samples and introducing Fisher-information regularization to prevent drift from the source model. We use Adam (learning rate 1×10^{-3}) with entropy margin $E_0 = 0.5$, diversity margin $d = 0.05$, and Fisher size 100.

Normalization/statistics baselines include DUA and DELTA. DUA performs gradient-free BN-statistics updates with exponentially decayed momentum over $T = 10$ adaptation steps: an internal state $\hat{\mu}_t = 0.9\,\hat{\mu}_{t-1}$ (initialized at $\hat{\mu}_0 = 0.1$) decays toward zero, and the actual BN momentum applied at each step is $\mu_t = \hat{\mu}_t + 0.05$, converging to the minimum momentum 0.05. To ensure stable estimation, each batch is augmented via input replication to obtain robust target-domain mean and variance. DELTA performs test-time BN renormalization by replacing standard BN layers with a re-normalization module that fuses current batch statistics with a strong source prior (weight 0.95); it

Table 13. **WiHAR-Dual OTTA Benchmark (Cross-Device Setting).** Comparison of *Online Test-Time Adaptation* (OTTA) methods for cross-device CSI sensing between Intel 5300 (D1) and Atheros (D2) devices. We report classification accuracy (%), FLOPs, trainable parameters, and inference latency.

Metric	Cross-Device	Base	BN Calib.		Entropy Min.		Pseudo-Labeling		Anti-Forget. Reg.		Consistency Reg.	
		Base	DUA	DELTA	TENT	SAR	T3A	TAST	CoTTA	EATA	RMT	PETAL
A. Effectiveness (Accuracy %)												
Accuracy	D1 → D1	99.20	**99.25**	99.10	99.20	99.01	97.27	87.93	99.06	98.77	96.37	99.15
	D1 → D2	15.51	**23.24**	21.64	22.11	20.46	21.55	20.79	20.37	20.27	19.99	20.74
	D2 → D2	98.49	**98.54**	97.88	97.41	97.45	92.69	84.54	97.27	97.03	94.72	97.83
	D2 → D1	14.19	15.89	22.07	22.44	22.58	**24.47**	18.06	22.07	20.79	17.40	22.35
B. Efficiency Metrics												
FLOPs (M)	All	559.94	1754.47	552.90	559.94	552.90	559.94	559.94	19597.82	559.94	2231.56	19597.82
Params	All	0	0	480	480	480	0	16768	4370000	480	12690000	4370000
Latency (ms)	All	0.05	0.11	0.36	0.13	0.98	0.30	0.43	23.05	0.31	0.58	6.41

Table 14. **WiHAR-Dual TTDA Benchmark (Cross-Device Setting).** Comparison of *Test-Time Domain Adaptation* (TTDA) methods for cross-device CSI sensing between Intel 5300 (D1) and Atheros (D2) devices. We report classification accuracy (%), FLOPs, trainable parameters, adaptation time, and inference latency.

Metric	Cross-Device	Base	Pseudo-Labeling			Consistency		Clustering		Source Est.		Self-Supervision
		Base	SHOT	ASL	BMD	APA	SFDA-UR	ASFA	ISFDA	BAIT	DaC	SHOT++
A. Effectiveness (Accuracy %)												
Accuracy	D1 → D1	99.20	99.20	98.16	98.87	99.29	99.39	99.29	99.29	99.15	99.34	**99.39**
	D1 → D2	15.51	26.97	**31.73**	25.79	22.49	22.30	29.37	21.45	22.40	26.40	27.96
	D2 → D2	98.49	98.49	95.00	97.12	**98.73**	98.59	98.49	98.44	98.30	98.59	98.40
	D2 → D1	14.19	14.19	26.31	28.62	22.35	21.68	**44.55**	7.92	23.29	23.81	43.56
B. Efficiency Metrics												
FLOPs (M)	All	559.94	559.94	559.94	559.94	552.90	559.94	559.94	552.90	559.94	559.94	559.94
Params (M)	All	0	0.27	4.37	4.10	4.37	4.38	4.37	4.37	4.37	4.37	4.37
Adaptation Time (s)	All	0	22.39	110.08	26.29	22.05	43.93	18.94	28.83	27.78	113.22	117.49

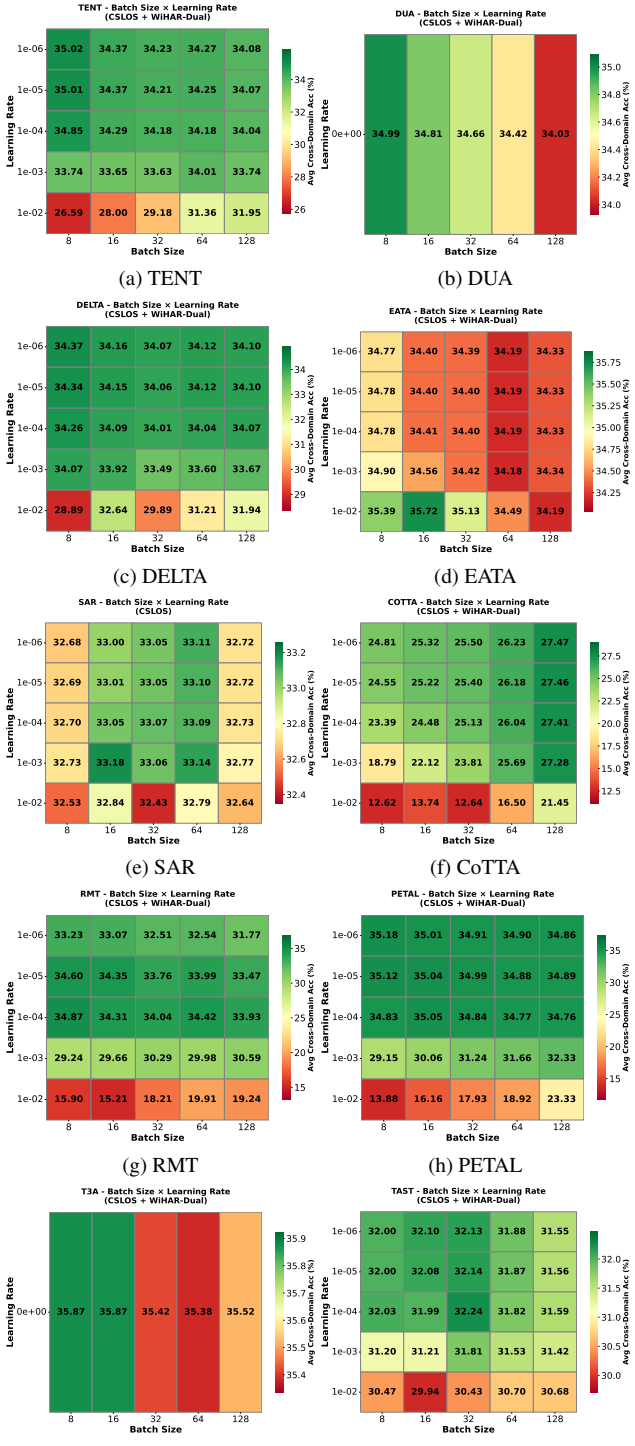
uses Adam (learning rate 1×10^{-6}) to minimize an entropy-based objective with class-balance EMA (decay 0.999) and entropy weighting enabled.

SAR employs Sharpness-Aware Minimization (SAM), which updates parameters using gradients computed at a worst-case perturbed point within radius $\rho = 0.05$. We use Adam as the base optimizer (learning rate 1×10^{-4}). SAR further filters unreliable samples by retaining those with entropy below $0.4 \log C$ (where C is the number of classes), and triggers model recovery when the moving-average entropy falls below $0.1 \log C$.

CoTTA and PETAL are not restricted to BN-only updates: both enable gradients throughout the model and optimize trainable parameters with Adam, updating the full network via mean-teacher consistency between the student prediction and an augmentation-averaged EMA teacher. CoTTA uses teacher EMA momentum $\alpha = 0.99$ and selects between the standard teacher and the augmentation-averaged teacher based on an anchor confidence threshold ($a_p = 0.85$). Stochastic restoration ($p_{\text{rst}} = 0.05$) is applied to prevent drift, and the learning rate is 1×10^{-7}. PETAL builds on this pipeline but additionally regularizes parameters with a self-penalization term (weight spw $= 1 \times 10^{-8}$) and restores low-importance parameters based on gradient-squared scores. For PETAL, we use learning rate 5×10^{-4}, $\alpha = 0.99$, $a_p = 0.9$, and $p_{\text{rst}} = 0.05$.

RMT updates all model parameters along with a learnable projection head (dimension 512) using Adam (learning rate 5×10^{-5}) under an EMA teacher with momentum 0.999. It enforces student–teacher consistency via a symmetric cross-entropy loss (weight $\lambda_{\text{ce}} = 15.0$) while simultaneously optimizing an augmentation-based contrastive objective on the projected features (weight $\lambda_{\text{cont}} = 0.5$, temperature $\tau = 0.7$). The weights λ_{ce} and λ_{cont} balance prediction alignment and representation invariance.

Finally, T3A is gradient-free (no optimizer): it maintains a per-class support set of pseudo-labeled features and adapts the classifier by recomputing class weights from the top-K lowest-entropy supports ($K = 36$), thereby suppressing uncertain samples without updating network weights. In contrast, TAST performs lightweight gradient-based adaptation by freezing the backbone and classifier and optimizing only a BatchEnsemble projection head with Adam (learning rate 1×10^{-4}). Specifically, test and support features are pro-



(a) TENT (b) DUA

(c) DELTA (d) EATA

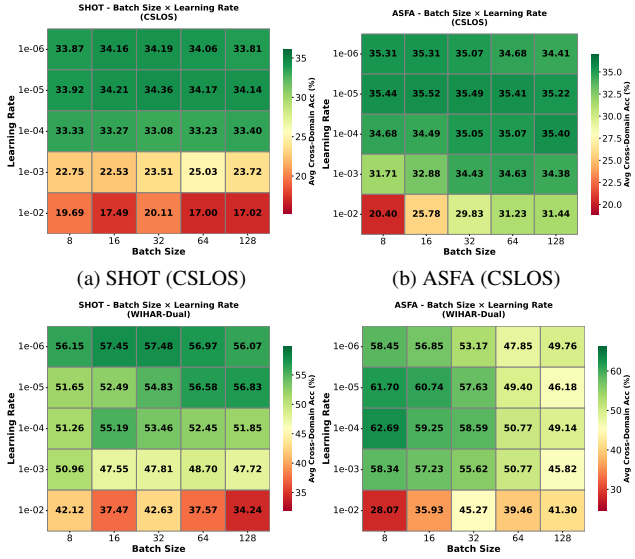
(e) SAR (f) CoTTA

(g) RMT (h) PETAL

(i) T3A (j) TAST

Figure 13. Hyperparameter sensitivity of representative OTTA methods averaged over WiHAR-Dual and CSLOS. Each heatmap shows mean cross-domain accuracy (%) under varying learning rates (x-axis) and update steps (y-axis).

jected into an ensemble feature space with 4 members. For each test sample, pseudo-targets are generated by performing k-NN voting ($k = 5$) over the top-K lowest-entropy



(a) SHOT (CSLOS) (b) ASFA (CSLOS)

(c) SHOT (WiHAR-Dual) (d) ASFA (WiHAR-Dual)

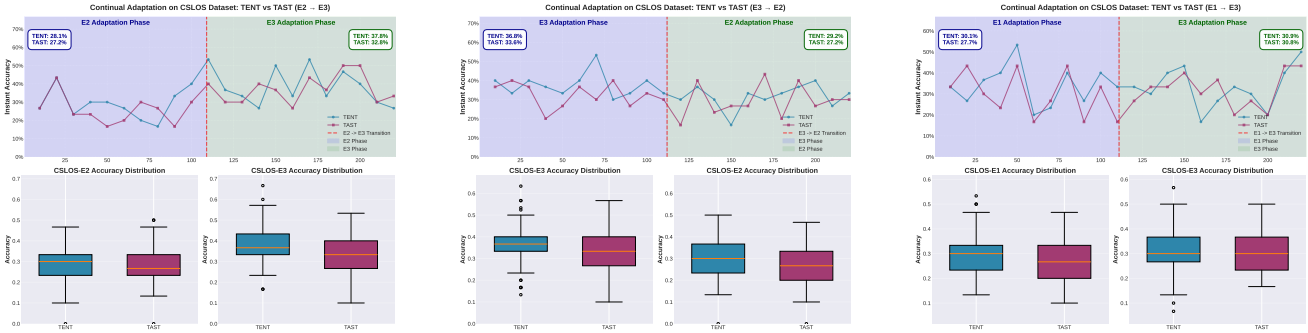
Figure 14. Hyperparameter sensitivity of representative TTDA methods on CSLOS and WiHAR-Dual datasets. Each heatmap shows accuracy (%) under varying learning rates (horizontal axis) and batch sizes (vertical axis).

support features per class ($K = 24$), using cosine similarity to measure neighbor proximity, and the projection head is updated for $\gamma = 2$ steps per batch to improve neighbor consistency. Finally, logits are scaled by temperature $\tau = 20.0$ to stabilize predictions.

Adaptation Details of TTDA. All TTDA methods access the entire unlabeled target dataset across multiple epochs with a shared batch size of 36. SHOT freezes the classifier head and adapts only the encoder for 50 epochs using Adam (learning rate 5×10^{-7}, weight decay 1×10^{-5}), optimizing an information-maximization objective together with pseudo-label cross-entropy weighted by $\lambda_{\text{cls}} = 0.1$. Pseudo-labels are refreshed every 50 iterations via centroid-based cosine assignment.

SHOT++ follows a two-stage pipeline. In Stage 1, only the encoder is adapted for 50 epochs using Adam (learning rate 5×10^{-7}, weight decay 1×10^{-5}), combining information maximization with an MSE consistency loss (weight 0.5) and a self-supervised shift-prediction loss (weight 0.5). Stage 2 performs MixMatch refinement for 10 epochs with MixUp ($\alpha = 0.75$), temperature sharpening ($T = 0.5$), and unsupervised loss weight $\lambda_u = 75$, optimizing both the encoder and classifier using SGD (learning rate 1×10^{-3}, momentum 0.9, weight decay 1×10^{-3}).

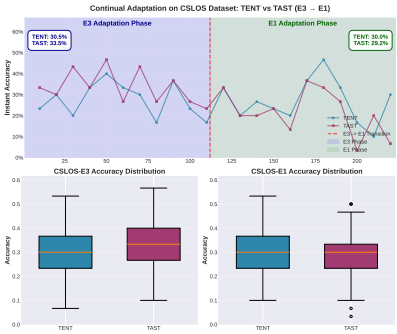
ASL alternates between (i) per-epoch class-balanced pseudo-label assignment via Sinkhorn–Knopp [4] and (ii) gradient-based adaptation for 20 epochs using Adam (learning rate 1×10^{-7}, weight decay $\times10^{-3}$). The objective combines pseudo-label cross-entropy (weight 0.1), entropy regularization (weight 0.2), and VAT-based robustness (weight 0.03). We apply mild noise and shift augmentation (Gaus-



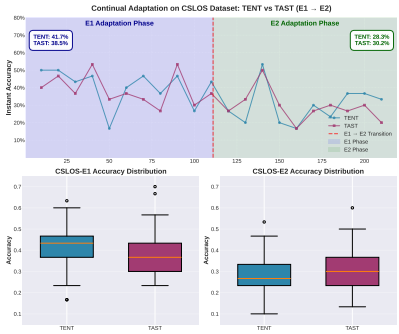
(a) CSLOS ($E1 \rightarrow E2 \rightarrow E3$)

(b) CSLOS ($E1 \rightarrow E3 \rightarrow E2$)

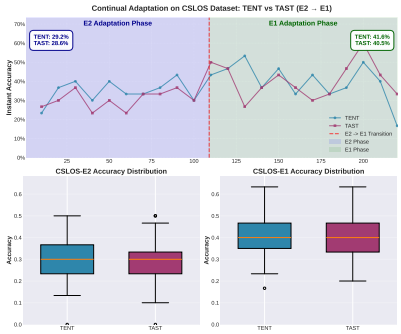
(c) CSLOS ($E2 \rightarrow E1 \rightarrow E3$)



(d) CSLOS ($E2 \rightarrow E3 \rightarrow E1$)



(e) CSLOS ($E3 \rightarrow E1 \rightarrow E2$)



(f) CSLOS ($E3 \rightarrow E2 \rightarrow E1$)

Figure 15. **CSLOS: Full continual adaptation trajectories and accuracy distributions across all 6 domain routes.** Each subfigure (top: adaptation curve, bottom: accuracy distribution) shows batch-wise adaptation accuracy and stability under sequential environment transfer.

sian noise $\sigma = 0.01$, vertical shifts up to 10 pixels and horizontal shifts up to 1 pixel).

BMD builds 4 prototypes per class in a 256-D embedding space using KMeans (50 iterations) initialized from top-confidence samples ($\texttt{topk} = 8$); it then adapts only the backbone BatchNorm layers and the embedding bottleneck (classifier and other backbone weights frozen) with SGD for 15 epochs (learning rate 1×10^{-3}), using pseudo-label supervision weighted by $\lambda_{\text{psd}} = 2.0$ and an additional prototype-similarity (soft-label) term weighted by $\lambda_{\text{dym}} = 0.5$.

ASFA updates the backbone using a loss that explicitly reduces inter-class confusion in the target-domain predictions (temperature $t_{\text{mcc}} = 3.0$, order 2.0) and a consistency term computed from $k_{\text{aux}} = 1$ augmented views with weight $\beta = 0.05$ (feature keep ratio in $[0, 1]$), trained with SGD (learning rate 1×10^{-5}) for 20 epochs.

ISFDA freezes the classifier head and adapts the backbone and bottleneck using SGD for 30 epochs (learning rate 1×10^{-5}). The objective combines pseudo-label cross-entropy ($\lambda_{\text{cls}} = 0.3$), entropy regularization with global diversity correction ($\lambda_{\text{ent}} = 1.0$), and a secondary-label correction term weighted by $\lambda_{\text{cls}}\lambda_{\text{scd}} = 0.3 \times 0.2$.

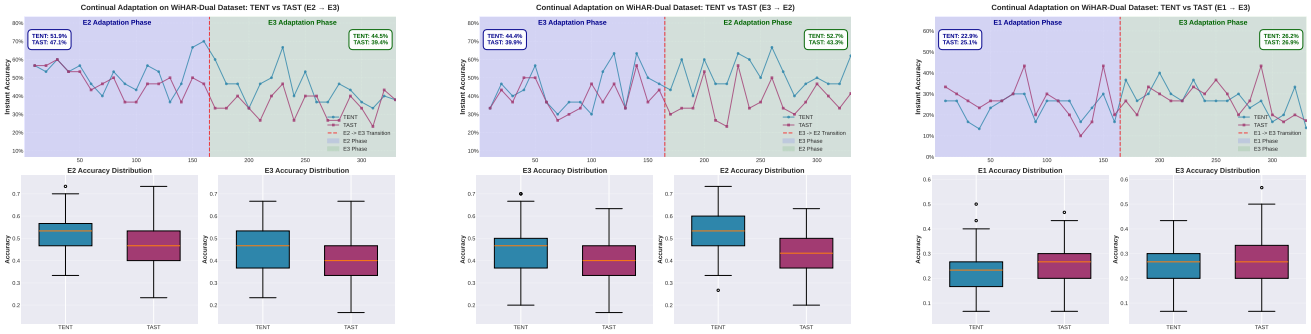
APA performs source-free adaptation by optimizing the model with VAT regularization in feature space, using Adam (learning rate 5×10^{-5}, weight decay 10^{-3}, batch size 36) with VAT (weight 0.03, $\epsilon = 20$, $\xi = 2$, 1 iteration) for 54 epochs.

SFDA-UR runs self-training in 3 rounds of 5 epochs each (learning rate 1×10^{-4}, batch size 36): at each round it updates pseudo-labels via class-balanced confidence thresholding, and then optimizes the model with information-maximization; uncertainty noise is modeled using auxiliary decoders with a consistency penalty.

BAIT aligns the adaptive classifier predictions to a frozen source classifier via a symmetric KL objective on top-entropy samples (CAST, weight 5.0, top-k ratio 0.4), together with class-balance regularization on the mean predictions (weight 0.5), optimized with SGD (learning rate 5×10^{-4}, momentum 0.9, weight decay 5×10^{-4}) for 50 epochs with module-wise LR multipliers (F: 0.005×, B: 0.05×, C: 0.05×).

DaC combines a momentum memory bank (momentum 0.5) with confident pseudo-label updates (threshold $p = 0.97$) and top-k neighbor voting ($k = 5$), trained with SGD (learning rate 10^{-7}) for 10 epochs. The total loss combines pseudo-label cross-entropy ($\lambda_{\text{cls}} = 0.3$), entropy regularization ($\lambda_{\text{ent}} = 1.0$), memory-bank neighbor contrastive loss ($\lambda_m = 0.5$), and an MMD adaptation term ($\lambda_{\text{ad}} = 0.2$).

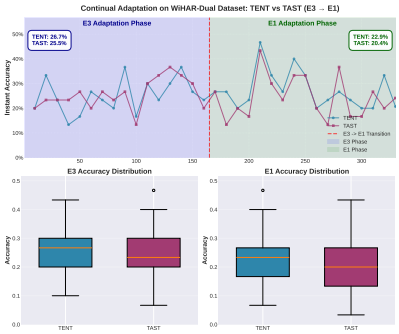
For methods that require stochastic perturbations to form multiple views (e.g., CoTTA, PETAL, RMT, and the random-augmentation interface in APA), we apply WiFi-appropriate CSI perturbations (e.g., mild noise, ampli-



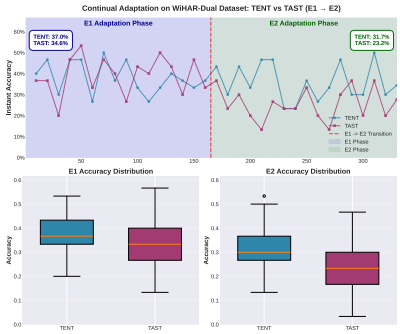
(a) WiHAR-Dual ($E1 \rightarrow E2 \rightarrow E3$)

(b) WiHAR-Dual ($E1 \rightarrow E3 \rightarrow E2$)

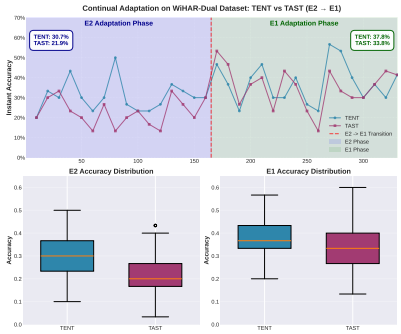
(c) WiHAR-Dual ($E2 \rightarrow E1 \rightarrow E3$)



(d) WiHAR-Dual ($E2 \rightarrow E3 \rightarrow E1$)



(e) WiHAR-Dual ($E3 \rightarrow E1 \rightarrow E2$)



(f) WiHAR-Dual ($E3 \rightarrow E2 \rightarrow E1$)

Figure 16. **WiHAR-Dual: Full continual adaptation trajectories and accuracy distributions across all 6 domain routes.** Each subfigure (top: adaptation curve, bottom: accuracy distribution) shows batch-wise adaptation accuracy and stability under sequential environment transfer.

tude/phase perturbations, and small affine-like distortions in the time–subcarrier plane). For SHOT++, the self-supervised stage uses temporal circular shift (cyclic-shift)[2] as a pretext task, where the model predicts the discrete shift level applied to each sample. For TTDA methods that require augmented views for contrastive or pseudo-label refinement (e.g., DaC and ASL), we apply mild noise injection and random spatial shifts. All experiments fix the random seed to 42. The full hyperparameter configurations for the 20 evaluated methods are released in our open-source repository.

D. Extended Benchmark Results

D.1. Feature-Space Visualization under Domain Shifts

Fig. 10 presents the extended feature-space visualizations across all CE, CS, and CD routes. Each subfigure plots PCA, t-SNE, and UMAP embeddings of the encoder's intermediate features from the pretrained source model. Blue and red points denote source and target samples, directly revealing the degree of feature overlap or separation across domain shifts.

Across the three CSLOS environments ($E1 \rightarrow E2$, $E1 \rightarrow E3$, and $E2 \rightarrow E3$), we observe the feature shifts mirror the underlying physical layouts. Transfers between LOS rooms, e.g., $E1 \rightarrow E2$ shows a moderate drift as features move from a compact lab room to a corridor with richer multipath. While LOS to NLOS transitions, e.g., $E1 \rightarrow E3$ exhibits the strongest deformation due to the NLOS wooden-partition setup, where attenuation, diffraction, and distorted multipath significantly alter feature envelopes. These transitions produce an increasingly stretched latent geometry while retaining a coarse global alignment.

In the CS routes (E1, E2, E3 intra-environment comparisons), the visualizations reveal greater internal reorganization within class manifolds. Although source and target features are globally aligned, the intra-cluster boundaries become blurred, and class cohesion weakens. This corresponds to human-specific variations, such as body morphology, gait dynamics, and movement speed, that perturb the temporal–spectral patterns of CSI while preserving overall topology. This suggests CS shifts primarily induce *local geometric perturbations*, not global statistical drift.

In contrast, the CD setting (Atheros → Intel and Intel → Atheros) show the most severe domain divergence. The two feature clouds become nearly disjoint, often mirrored

[2]We further verified that other common signal augmentations, i.e., time warping, masking, dropout, noise. Generally, cyclic-shift offers comparable performance while maintaining protocol uniformity across all baselines.

or rotated in low-dimensional embeddings. This reflects strong hardware-induced distortions caused by antenna calibration bias, carrier frequency offset (CFO), and I/Q imbalance. Even under identical environments and activities, device heterogeneity leads to fundamental manifold misalignment that cannot be compensated by simple normalization.

Overall, the extended visualizations in Fig. 10 show the hierarchical nature of domain shifts in WiFi sensing: mild statistical drift in CE, localized geometric perturbations in CS, and near-complete manifold disjunctions in CD. These patterns offer a physically interpretable view of how environmental layouts, human variations, and hardware differences reshape CSI representations, motivating robust TTA methods dedicated to wireless sensing.

D.2. Class-Level Feature Inconsistency at Test Time

Fig. 11 and Fig. 12 present the extended class-wise t-SNE visualizations across all CE, CS, and CD configurations in CSLOS and WiHAR-Dual. Each subfigure shows the predicted target-domain embeddings from the pretrained model, where colors denote activity categories.

For WiHAR-Dual (Cross-Environment, Cross-Device), the degree of separability aligns closely with the physical similarity of environments. Transfers between semantically close rooms (e.g., $E1 \rightarrow E2$, seminar room to lounge space), clusters remain compact and separated, showing well-preserved inter-class boundaries. As the environmental gap widens ($E1 \rightarrow E3$ or $E2 \rightarrow E3$, from seminar/lounge room to exhibition space), neighboring clusters begin to merge, and the feature manifold deforms gradually, exhibiting irregular boundaries and scattered intra-class densities. Consequently, larger environmental gaps (e.g., involving exhibition space) result in the most entangled embeddings, where class topology becomes highly mixed. In contrast, Cross-device transfers ($D1 \rightarrow D2$, $D2 \rightarrow D1$) yield almost disjoint manifolds, reflecting severe calibration bias and device-dependent representation gaps.

For CSLOS (Cross-Subject, Cross-Environment), the trends are more pronounced. Cross-subject settings show moderate manifold reorganization, clusters remain roughly aligned but become internally distorted due to individual motion variations. Cross-environment routes introduce both statistical displacement and geometric deformation, producing increasingly fragmented structures.

The evolution across all settings suggest a transition pattern: *compact* $\rightarrow$ *fragmented* $\rightarrow$ *entangled*, capturing how statistical displacement under mild gaps evolves into geometric deformation and ultimately manifold breakdown as the domain discrepancy grows.

D.3. Measure Domain Discrepancy via MMD

To complement the clustering-based metrics reported in the main text, we compute MMD [11] at both the feature level (penultimate-layer embeddings) and the BatchNorm-statistic level (running mean and variance) across environment pairs on WiHAR-Dual. Results are summarized in Table 15.

Table 15. **Feature- and BN-statistic-level MMD** across environment pairs on WiHAR-Dual.

Metric	$(E1, E2)$	$(E1, E3)$	$(E2, E3)$
Feature MMD	0.213	0.051	0.402
BN-MMD	38.15	11.01	54.94

Feature-level MMD exhibits only a weak partial ordering across domains: $\text{MMD}(E2, E3) > \text{MMD}(E1, E2) \gg \text{MMD}(E1, E3)$. Notably, the $(E1, E3)$ pair, which differs substantially in room size and multipath density, paradoxically shows the *lowest* feature MMD. BN-statistic MMD is even more unstable: the $(E2, E3)$ pair shows the largest discrepancy (54.94), yet this ordering still does not align with the observed ranking of TTA performance.

Overall, neither geometric metrics (SS, CH, DB) nor distributional discrepancy metrics (MMD) reliably predict TTA difficulty in WiFi sensing. This likely arises from complex nonlinear distortions induced by latent physical factors (e.g., propagation geometry and hardware calibration), which cannot be captured by simple pairwise distance measures.

D.4. WiTTA-Benchmarking Result Tables

Full comparison results for all evaluated TTA methods are shown in Table 7 to 14. Each table reports classification accuracy (%), FLOPs, trainable parameters, training or adaptation time, and inference latency, following the same settings as §C.

Across all CE, CS, and CD settings on WiHAR-Dual and CSLOS, there is a consistent hierarchy of difficulty: CE $<$ CS $<$ CD, with CD showing the biggest hardness and requiring the strongest adaptation.

For OTTA, lightweight normalization/entropy–driven methods (e.g., DUA, DELTA, TENT, EATA) deliver stable, moderate gains at low cost, while heavier consistency-based methods (e.g., CoTTA, PETAL, RMT) offer higher accuracy but with large FLOPs/latency.

For TTDA, pseudo-labeling and clustering families achieve substantial accuracy boosts across all CE/CS/CD routes, far surpassing OTTA, particularly ASFA, SHOT, and BMD, which yield the largest improvements (up to $+30$ pp absolute accuracy gains over Base) under severe environment or device shifts (e.g., $D2 \rightarrow D1$). However, these methods incur significant parameter overhead and long adaptation times, revealing a clear accuracy–cost trade-off.

Overall, OTTA offers stable and low-cost but modest improvements, whereas TTDA delivers larger yet high-cost gains. Among the three domain shifts, CD is the hardest one, where only strong TTDA methods can meaningfully bridge the gap.

E. Extended Experiments on Factors Influencing TTA Effectiveness

E.1. Hyperparameter Sensitivity

To further investigate the stability and robustness of TTA algorithms under varying hyperparameters, we present a detailed sensitivity analysis across *batch size* (horizontal axis, range$[8, 128]$) and *learning rate* (vertical axis, range $[10^{-6}, 10^{-2}]$) in Figs. 13 and 14. Each heatmap illustrates the average accuracy (%) over *CSLOS* and *WiHAR-Dual* datasets, where greener shades correspond to higher performance.

OTTA (Fig. 13). Normalization-/entropy-based OTTA methods (TENT, DUA, DELTA, EATA) show *broad and flat* accuracy trends over learning rates 10^{-6}-10^{-3}, showing remarkable insensitivity to hyperparameter variation. This robustness arises from OTTA's *local, low-magnitude updates* (e.g., BN-statistics recalibration or entropy minimization) that adjust only feature statistics rather than model weights. Such an update mechanism acts as an implicit regularizer that preserves pre-trained representations and reduces gradient noise, enabling stable adaptation even under noisy CSI signals. Consistency-based methods (CoTTA, RMT, PETAL) raise quickly to peak accuracy since they rely on stochastic restoration and self-ensembling, yet they still maintain relatively smooth landscapes compared to TTDA. OTTA's stability stems from its lightweight, self-corrective update mechanism that aligns target statistics without propagating pseudo-label errors.

TTDA (Fig. 14). In contrast, TTDA methods conduct *explicit re-training* on unlabeled targets using pseudo-labels or clustering, where both the loss and gradient magnitudes are strongly influenced by hyperparameters. Pseudo-label fine-tuning model (e.g., SHOT) remains relatively stable because it freezes classifier head and adapts only the feature extractor. However, clustering-based methods such as ASFA show a high-gain peak: their accuracy quickly collapses when the learning rate or batch size deviates from the peak (e.g., 62.7% $\rightarrow$ 28% on WiHAR-Dual). This sensitivity originates from recursive pseudo-label propagation and cluster assignment drift. This makes small early errors in pseudo-labels amplify across iterations, leading to unstable feedback loops. Consequently, TTDA achieves larger adaptation gains at the cost of fragile convergence, whereas OTTA trades adaptivity for stability.

Takeaway. Comparing OTTA and TTDA, normalization- and entropy-based OTTA methods (e.g., TENT, DUA, EATA) achieve an average mean accuracy of 34.5% with a very small deviation of $\pm$0.4, corresponding to a stability index roughly 2.5$\times$ higher than that of TTDA. In contrast, TTDA approaches (e.g., SHOT and ASFA) reach a higher mean accuracy of 41.8% but has much larger variance $\pm$2.3, indicating weaker robustness to hyperparameter drift.

For streaming WiFi sensing, OTTA is thus preferred for online deployment, whereas TTDA is more suitable for offline re-calibration or cross-device adaptation where tuning is feasible.

E.2. Continual Adaptation Analysis

Unlike static vision tasks, real-world WiFi sensing environments are inherently *dynamic*: TX and RX may be relocated, antenna orientations adjusted, or environment layout changed over time. To assess whether TTA methods can remain robust under such non-stationary conditions, we introduce a *continual adaptation evaluation*, where models sequentially adapt environments (e.g., $E1 \rightarrow E2 \rightarrow E3$), simulating long-term and dynamic WiFi sensing scenarios. Figs. 15 and 16 show cumulative accuracy trajectories (top) and accuracy distributions (bottom) for representative OTTA methods, i.e., TENT (entropy minimization) and TAST (pseudo-labeling), under different continual adaptation routes on *CSLOS* and *WiHAR-Dual*.

Adaptation Stability. Across both datasets, the two OTTA methods generally remain stable under continual distribution shifts. The cumulative accuracy of the normalization-based model TENT is smoother. TENT acts like low-pass filters that absorb slow statistical changes. In contrast, pseudo-labeling model TAST behaves as a high-gain amplifier that achieves higher peaks at the expense of stability, especially at the second hop (e.g., CSLOS $E2 \rightarrow E3 \rightarrow E1$).

Route-Sensitivity. Continual adaptation is highly *route-sensitive*. When models start from the same environment but follow different routes, their results diverge notably: on CSLOS, $E1 \rightarrow E2 \rightarrow E3$ (37.8%) outperforms $E1 \rightarrow E3 \rightarrow E2$ (29.2%) by +8.6 pp, showing that early adaptation direction shapes later normalization statistics. Conversely, when the final domain is fixed, different routes remain different, e.g., CSLOS E1 ends at 41.6% for $E3 \rightarrow E2 \rightarrow E1$ versus 30% for $E2 \rightarrow E3 \rightarrow E1$ (+11.6 pp). This suggests that adaptation is *non-commutative*: intermediate domains accumulate distinct statistics and pseudo-label biases that continue to affect later domains.

Dataset-Dependency. CSLOS exhibits relatively smooth transitions (3–5 pp variance) because its environments are compact and structurally similar, producing milder physical drift. In contrast, WiHAR-Dual shows much stronger fluctuations (10–20 pp) even though the same device is used. WiHAR-Dual's room geometries and

Table 16. **WiHAR-Dual OTTA Backbone Results (Cross-Environment Setting, MobileNetV2).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Environment (CE)** protocol using the MobileNetV2 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	Source	Target	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
			Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)							
Accuracy	E1	E1	**96.67**	96.04	95.55	95.05	94.00
	E2	E1	**27.40**	25.14	19.88	24.00	25.16
	E3	E1	22.99	26.19	24.99	26.27	**27.56**
	E1	E2	32.75	**54.58**	53.30	44.07	47.06
	E2	E2	95.35	98.28	**98.46**	97.51	96.44
	E3	E2	15.11	42.94	42.35	**43.00**	41.22
	E1	E3	22.51	32.01	26.75	31.48	**32.47**
	E2	E3	28.67	38.63	40.86	42.09	**43.08**
	E3	E3	84.34	**89.65**	83.51	87.59	85.01
B. Efficiency Metrics							
FLOPs (M)	All	All	5860.11	5860.11	5860.11	5862.54	5860.11
Params (K)	All	All	0	34.11	34.11	0	34.11

Table 17. **WiHAR-Dual OTTA Backbone Results (Cross-Environment Setting, ResNet-10).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Environment (CE)** protocol using the ResNet-10 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	Source	Target	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
			Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)							
Accuracy	E1	E1	**94.18**	92.58	92.79	88.32	92.81
	E2	E1	25.38	27.68	28.73	**30.09**	29.04
	E3	E1	20.27	32.77	33.28	28.47	**33.40**
	E1	E2	46.55	48.03	48.47	43.36	**48.82**
	E2	E2	**97.80**	96.63	97.09	94.71	97.15
	E3	E2	16.59	45.54	44.76	**52.86**	44.88
	E1	E3	18.12	36.53	**36.73**	35.30	36.41
	E2	E3	27.42	45.67	46.41	43.48	**46.66**
	E3	E3	98.52	**98.53**	98.10	97.39	98.10
B. Efficiency Metrics							
FLOPs (M)	All	All	14029.16	14029.16	14029.16	14029.16	14029.16
Params (K)	All	All	0	5.87	5.87	0	5.87

multipath conditions differ more substantially, amplifying distribution shifts during sequential adaptation.

Forgetting across Sequential Hops. The second domain hop typically causes *mild catastrophic forgetting*, particularly for TAST (e.g., CSLOS $E3 \rightarrow E1 \rightarrow E2$, WiHAR-Dual $E1 \rightarrow E2 \rightarrow E3$), as previously calibrated features are overwritten before new ones.

Takeaway. Continual Wi-TTA reveals a dual nature: it is strongly *route-sensitive* and *dataset-dependent*. These behaviors are caused by the interplay between accumulated feature-statistics drift and the specific physical complexity of each environment. This highlights the need for models that judiciously choose adaptation routes and remain robust under evolving wireless conditions.

E.3. Backbone Generality Analysis

To evaluate whether the key findings of WiTTA-Bench are backbone-dependent, we replace the default four-block CNN encoder with MobileNetV2 [36] and ResNet-10 [14]as the feature extraction backbone networks to cover two typical CNN types: lightweight mobile architecture and residual architecture. MobileNetV2 has lower computational and parameter requirements, making it suitable for online inference in OTTA scenarios; ResNet10 has more regular residual modules and sufficient Batch Normalization layers, making it structurally stable and commonly

Table 18. **CSLOS OTTA Backbone Results (Cross-Subject Setting, MobileNetV2).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Subject (CS)** protocol using the MobileNetV2 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	**Cross-Subject**	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
		Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)						
Accuracy	E1	**97.34**	96.39	96.29	97.34	93.68
	E1 cross-subject	28.82	32.28	**35.21**	26.34	34.91
	E2	**97.45**	97.25	96.79	97.14	94.70
	E2 cross-subject	37.54	40.89	**41.98**	37.15	41.28
	E3	96.36	95.96	95.81	**96.46**	91.42
	E3 cross-subject	37.56	36.74	35.26	36.67	**37.56**
B. Efficiency Metrics						
FLOPs (M)	All	2987.78	2987.78	2987.78	2987.78	2987.78
Params (K)	All	0	34.11	34.11	0	34.11

Table 19. **CSLOS OTTA Backbone Results (Cross-Subject Setting, ResNet-10).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods under the **Cross-Subject (CS)** protocol using the ResNet-10 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	**Cross-Subject**	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
		Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)						
Accuracy	E1	90.97	**95.08**	93.33	83.04	93.38
	E1 cross-subject	29.50	33.63	33.71	23.70	**34.46**
	E2	**90.06**	88.93	89.04	82.05	89.19
	E2 cross-subject	39.56	43.61	43.93	38.40	**44.70**
	E3	94.16	**95.96**	94.91	89.98	95.26
	E3 cross-subject	35.26	**40.22**	38.96	34.74	39.41
B. Efficiency Metrics						
FLOPs (M)	All	7154.69	7154.69	7154.69	7154.69	7154.69
Params (K)	All	0	5.87	5.87	0	5.87

used for adaptive research under distribution shifts.

Transformer architectures are not chosen because most TTA methods evaluated in WiTTA-Bench (e.g., TENT, EATA, SAR, ASFA) are BN-centric: they adapt by modifying BatchNorm statistics or affine parameters. Transformer architectures typically use LayerNorm instead of BatchNorm, making the direct application of these methods not straightforward.

During training of the MobileNetV2 and ResNet-10 base models, hyperparameters vary across settings. In the cross-environment setting, MobileNetV2 is optimized with Adam (learning rate 1×10^{-3}, weight decay 1×10^{-4}) for 100 epochs with batch size 32, while ResNet-10 uses a lower learning rate 1×10^{-5} with the same weight decay for 100 epochs (batch size 30). In the cross-person setting, both models use Adam (learning rate 1×10^{-4}, weight decay 1×10^{-4}, batch size 30); MobileNetV2 is trained for 70 epochs with classifier dropout $p = 0.5$, while ResNet-10 is trained for 45 epochs. In the cross-device setting, both models use Adam (learning rate $5{\times}10^{-5}$, weight decay $1{\times}10^{-4}$, batch size 30) and are trained for 50 epochs.

The corresponding evaluation results for each configuration are presented across six separate tables (Tables 16–21), covering the CE, CS, and CD benchmarks under both MobileNetV2 and ResNet-10 architectures, respectively.

From the full results, the difficulty hierarchy appears largely backbone-invariant. CD remains the hardest setting in terms of source-model accuracy, and the Base (no-adaptation) results consistently show the lowest accuracy on CD routes, confirming that cross-device shifts are intrinsically the most severe. However, the effectiveness of TTA is backbone-dependent. After adaptation, ResNet-10 largely preserves the ordering CE $>$ CD, whereas MobileNetV2 achieves unexpectedly strong gains on CD (e.g., TENT: 39.35% on CD vs. 36.58% on CE), narrowing or even reversing the gap. Overall, despite substantial route-level variation, the broad ranking CE $<$ CS $<$ CD holds for both MobileNetV2 and ResNet-10, suggesting that the hierarchy reflects intrinsic physical properties of the domain shift rather than architecture-specific artifacts.

Table 20. **WiHAR-Dual OTTA Backbone Results (Cross-Device Setting, MobileNetV2).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods for cross-device CSI sensing between Intel 5300 (D1) and Atheros (D2) devices using the MobileNetV2 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	Cross-Device	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
		Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)						
Accuracy	D1 → D1	**95.66**	92.50	90.52	90.81	87.22
	D1 → D2	14.29	**42.39**	41.40	14.30	41.16
	D2 → D2	**95.62**	92.40	90.90	93.78	87.98
	D2 → D1	16.83	**36.30**	36.26	16.41	35.93
B. Efficiency Metrics						
FLOPs (M)	All	5860.11	5860.11	5860.11	5862.64	5860.11
Params (K)	All	0	34.11	34.11	0	34.11

Table 21. **WiHAR-Dual OTTA Backbone Results (Cross-Device Setting, ResNet-10).** Comparison of representative *Online Test-Time Adaptation* (OTTA) methods for cross-device CSI sensing between Intel 5300 (D1) and Atheros (D2) devices using the ResNet-10 backbone. We report classification accuracy (%), FLOPs, and trainable parameters.

Metric	Cross-Device	Base	Entropy Min.		Pseudo-Labeling	Anti-Forget. Reg.
		Base	TENT	SAR	T3A	EATA
A. Effectiveness (Accuracy %)						
Accuracy	D1 → D1	**96.75**	95.62	95.10	91.51	95.29
	D1 → D2	14.29	33.80	**34.75**	14.29	34.28
	D2 → D2	**93.31**	92.79	91.98	90.57	91.80
	D2 → D1	14.29	23.29	23.28	14.29	**23.57**
B. Efficiency Metrics						
FLOPs (M)	All	14029.16	14029.16	14029.16	14029.16	14029.16
Params (K)	All	0	5.87	5.87	0	5.87

References for Appendix

[1] Alsaify Baha'A, Mahmoud M Almazari, Rami Alazrai, and Mohammad I Daoud. A dataset for wi-fi-based human activity recognition in line-of-sight and non-line-of-sight indoor environments. *Data in Brief*, 33:106534, 2020.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

[3] Dhanajit Brahma and Piyush Rai. A probabilistic framework for lifelong test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3582–3591, 2023.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[5] Chen Chen, Gang Zhou, and Youfang Lin. Cross-domain wifi sensing with channel state information: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023.

[6] Michal Danilowski, Soumyajit Chatterjee, and Abhirup Ghosh. Botta: Benchmarking on-device test time adaptation. *arXiv preprint arXiv:2504.10149*, 2025.

[7] Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7704–7714, 2023.

[8] Qi Dou, Daniel Ching de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6450–6461, 2019.

[9] Chen Du, Yufei Wang, Jingwen Guo, Yifan Han, Jing Zhou, and Gao Huang. Unitta: Unified benchmark and versatile framework towards realistic test-time adaptation. *arXiv preprint arXiv:2407.20080*, 2024.

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[12] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[13] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM computer communication review*, 41(1):53–53, 2011.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[15] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[16] Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. In *International Conference on Learning Representations (ICLR)*, 2023.

[17] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9619–9628, 2021.

[18] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-stream convolution augmented transformer for human activity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):286–293, 2021.

[19] Bing Li, Wei Cui, Le Zhang, Ce Zhu, Wei Wang, Ivor W Tsang, and Joey Tianyi Zhou. Diffformer: Multi-resolutional differencing transformer with dynamic ranging for time series analysis. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):13586–13598, 2023.

[20] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy Hospedales. Episodic training for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6256–6263, Honolulu, HI, USA, 2019.

[21] Xinhao Li, Jingjing Li, Lei Zhu, Guoqing Wang, and Zi Huang. Imbalanced source-free domain adaptation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 3330–3339, 2021.

[22] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

[23] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.

[24] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34: 21808–21820, 2021.

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 97–105, Lille, France, 2015.

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2208–2217, Sydney, Australia, 2017.

[27] Yongsen Ma, Gang Zhou, and Shuangquan Wang. Wifi sensing with channel state information: A survey. *ACM Computing Surveys*, 52(3):1–36, 2019.

[28] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6568–6577, Long Beach, CA, USA, 2019.

[29] Wei Meng, Zhicong Liu, Bing Li, Wei Cui, Joey Tianyi Zhou, and Le Zhang. Graphar: A lightweight human activity recognition model by exploring the sub-carrier correlations. *IEEE Transactions on Wireless Communications*, 23 (4):2755–2770, 2024.

[30] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14765–14775, 2022.

[31] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5715–5725, Venice, Italy, 2017.

[32] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning (ICML)*, 2022.

[33] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

[34] S Prabhu Teja and Francois Fleuret. Uncertainty reduction for model adaptation in semantic segmentation. In *CVPR*, page 7, 2021.

[35] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *European conference on computer vision*, pages 165–182. Springer, 2022.

[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[37] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[38] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Subhajit Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[39] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. Magicol: Indoor localization using pervasive magnetic field and opportunistic wifi sensing.

IEEE Journal on Selected Areas in Communications, 33(7): 1443–1457, 2015.

[40] Julian Strohmayer, Rafael Sterzinger, Matthias Wödlinger, and Martin Kampel. Datta: Domain-adversarial test-time adaptation for cross-domain wifi-based human activity recognition. *arXiv preprint arXiv:2411.13284*, 2024.

[41] Tao Sun, Cheng Lu, and Haibin Ling. Domain adaptation with adversarial training on penultimate activations. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9935–9943, 2023.

[42] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[43] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, Honolulu, HI, USA, 2017.

[44] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[45] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2023.

[46] Kailong Wang, Cong Shi, Jerry Cheng, Yan Wang, Minge Xie, and Yingying Chen. Solving the wifi sensing dilemma in reality leveraging conformal prediction. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2023.

[47] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[48] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. Deepfi: Deep learning for indoor fingerprinting using channel state information. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1666–1671, 2015.

[49] Kun Xia, Lingfei Deng, Wlodzislaw Duch, and Dongrui Wu. Privacy-preserving domain adaptation for motor imagery-based brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, 69(11):3365–3376, 2022.

[50] Jiahao Xie, Zhenfeng Li, Chao Feng, Jingzhi Lin, and Xianjia Meng. Wi-am: Enabling cross-domain gesture recognition with commodity wi-fi. *Sensors*, 24(5):1354, 2024.

[51] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 53–64, 2015.

[52] Hao Yan, Yuhong Guo, and Chunsheng Yang. Augmented self-labeling for source-free unsupervised domain adaptation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[53] Jianfei Yang, Xinyan Chen, Dazhuo Wang, Han Zou, Chris Xiaoxuan Lu, Sumei Sun, and Lihua Xie. Sensefi: A library and benchmark on deep-learning-empowered wifi human sensing. *Patterns*, 4(3):100703, 2023.

[54] Shiqi Yang, Yaxing Wang, Luis Herranz, Shangling Jui, and Joost van de Weijer. Casting a bait for offline and online source-free domain adaptation. *Computer Vision and Image Understanding*, 234:103747, 2023.

[55] Guolin Yin, Junqing Zhang, Guanxiong Shen, and Yingying Chen. Fewsense: Towards a scalable and cross-domain wi-fi sensing system using few-shot learning. *IEEE Transactions on Mobile Computing*, 23(1):453–468, 2024.

[56] Jin Zhang, Bo Wei, Wen Hu, and Salil S. Kanhere. Wifi-id: Human identification using wifi signal. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 75–82, 2016.

[57] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. Crosssense: Towards cross-site and large-scale wifi sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 305–320, 2018.

[58] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Widar3. 0: Zero-effort cross-domain gesture recognition with wi-fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8671–8688, 2021.

[59] Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. *Advances in Neural Information Processing Systems*, 35:5137–5149, 2022.

[60] Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: Degradation-free fully test-time adaptation. In *International Conference on Learning Representations (ICLR)*, 2023.

[61] Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In *International Conference on Machine Learning*, pages 42058–42080. PMLR, 2023.

[62] Aihua Zheng, Zhihao Fei, Yuhe Ding, Chenglong Li, and Bin Luo. Ruler: Source-free domain adaptive person re-identification via uncertain label refinery. *Machine Intelligence Research*, 22(5):900–916, 2025.

[63] Naiyu Zheng, Yuanchun Li, Shiqi Jiang, Yuanzhe Li, Rongchun Yao, Chuchu Dong, Ting Chen, Yubo Yang, Zhimeng Yin, and Yunxin Liu. Adawifi: Collaborative wifi sensing for cross-environment adaptation. *IEEE Transactions on Mobile Computing*, 24(2):845–858, 2025.

[64] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[65] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.