DiMTE: Directed Model-Transfer Estimation for Wireless Datasets

Anonymous authors
Paper under double-blind review

Abstract

Many data decisions depend on how well a model trained on one dataset would perform on another, such as which dataset to train on or which to add to scarce labels. Measuring this usually means training and testing one model per dataset. Dataset similarity measures are cheaper, but they are often symmetric and none returns the downstream loss. We introduce **DiMTE**, a directed transfer-risk estimator built from *source-induced proxy predictors*. DiMTE embeds every sample with a shared frozen encoder, such as a foundation model, fits a lightweight class-conditional proxy to each source, and evaluates it on labeled target samples under the deployment loss. It thus estimates how a model trained on each source would perform on the target, in the deployment metric. We prove a uniform bound that separates finite-sample error from the mismatch between proxy and deployment model. Across city-disjoint wireless benchmarks, including cities ray traced with an independent simulator after the method was frozen, DiMTE ranks sources best among methods that train no deployment model per source, and remains useful with few target labels. Estimators that inspect a trained model per source can be stronger, but only once those models exist. The same estimator also ranks camera-trap locations and source languages, with no wireless prior. DiMTE therefore lets a practitioner rank a whole library of datasets in the deployment metric before any per-source model is trained.

1 Introduction

Modern ML systems often hold a library of labeled datasets from different environments. For a new target they need the library's transfer profile, the loss that a model trained on each dataset would reach there. The profile identifies the best source, its close alternatives and the candidates worth keeping. Measuring it directly means training and evaluating one deployment model per source, the computation that an estimate of the profile should avoid (Figure 1).

Wireless networks make the problem concrete. A base station, or site, serves the users of one area, and a sample is the radio channel between its antennas and one user. A site's channels reflect its buildings, propagation paths and user distribution, so a model trained at one site may degrade at another (Nguyen et al., 2022). Labels are costly, since finding a user's best beam means measuring every beam (3GPP, 2023). Operators therefore reuse labeled datasets from past sites. We study two tasks that 3GPP, the cellular standards body, studies for machine learning: (i) line-of-sight (LOS)/non-line-of-sight (NLOS) identification, a binary classification of whether buildings block a user's direct path, and (ii) spatial-domain downlink beam prediction, a 32-class choice of the transmit direction, or beam, that serves the user best. Our datasets come from ray tracing, which simulates radio propagation in 3D city models. They cover training and test cities of the public DeepMIMO collection (Alkhateeb, 2019) and new test cities that we ray traced with Sionna RT (Aït Aoudia et al., 2025) after the method was frozen. Appendix S defines every wireless and method term.

What a site needs is not how similar two datasets are, but how well a model trained on one performs on the other. Let h_s be the deployment model that a specified training procedure produces on source D_s. The quantity of interest is the risk of h_s on target D_t. That risk is *task dependent*. Two sites can share the structure needed for LOS/NLOS identification while requiring different beam-prediction boundaries. The risk is also *directed*, since a diverse source can cover a narrow target even when the reverse transfer fails (Figure 1c). A symmetric score cannot represent both transfer directions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

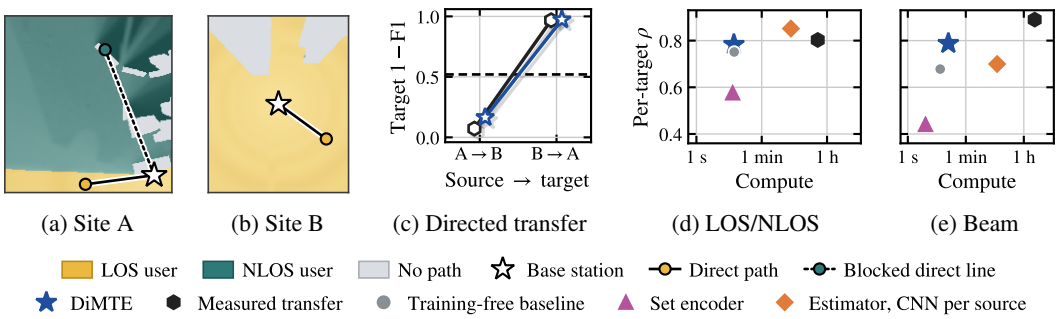


Figure 1: **Transfer is directed, and DiMTE estimates it before candidate deployment models are trained.** (a,b) Two sites of the DeepMIMO test cities, where brighter shades mean a stronger received signal. A user is line of sight (LOS) when its direct path to the base station is clear (solid line) and non-line of sight (NLOS) when buildings block it (dotted line), so only reflected or scattered paths arrive. Site A has mostly NLOS users and site B only LOS users, so a model trained on B never sees an NLOS user. (c) Models trained on A reach a 1−F1 of 0.07 on B and those trained on B 0.97 on A, and DiMTE estimates 0.17 and 0.97. The dashed line is their mean, the best a symmetric score can give. (d,e) Mean per-target ρ on the Sionna test cities against the compute to rank a new library, for the best method of each kind (Section 5.2), with *measured transfer* from one training seed. Every point assumes the task's one-off training-city matrix (1.4 and 3.6 GPU-hours), which sets DiMTE's three hyperparameters and trains the set encoder.

No existing approach estimates this directed, task-dependent risk before training. Dataset similarity measures need no deployment model, but they are usually symmetric and return a dissimilarity rather than a task loss. Transferability estimators such as LEEP, NCE and LogME are directed, but they inspect a trained model for every candidate (Nguyen et al., 2020; Tran et al., 2019; You et al., 2021). A library usually stores datasets rather than models, and a new task or model family would require training all of them again. Set encoders that learn to predict transfer need the measured transfer between every pair of past datasets, a transfer matrix. We seek a directed estimate in the task loss that is available before any candidate deployment model is trained.

Our key observation is that estimating the profile does not require training h_s. A cheap classifier built from the source is enough, as long as its loss on the target tracks that of h_s. We therefore introduce **DiMTE (Directed Model-Transfer Estimation)**, a source-induced proxy-risk estimator. A shared frozen encoder, such as a pretrained foundation model, maps every example into a common representation, the labeled embeddings of source D_s fit a lightweight proxy g_s, and labeled samples of the target evaluate g_s under the deployment loss. The source defines the proxy and the target its evaluation distribution, so the estimate $\widehat{P}_{s\to t}$ is directed and carries task units. DiMTE trains no candidate deployment model, so scoring a new target against an entire library costs one pass over the compact summary stored for each source. Our default proxy summarizes each source class by at most M prototypes and classifies a sample with a Gaussian mixture over them, a mixture discriminant rule. Nearest-neighbor and Parzen classifiers built from the source rank sources about as well, while one prototype per class or a linear classifier ranks worse. What matters is therefore a flexible classifier built from the source, not the Gaussian form.

We make three contributions.

- **A directed estimand in task units.** We define transfer for a specified training procedure and estimate the target risk of the model it produces by evaluating a lightweight source-induced proxy on labeled target samples. One proxy serves every deployment model, so we measure how closely it tracks each of two deployment models. The estimate carries the units of the deployment loss and needs no model trained on a candidate source, which separates the problem from symmetric dataset similarity and from selection among trained models.
- **A lightweight estimator and a diagnostic of its error.** A class-conditional prototype mixture provides a proxy whose storage does not grow with the source, and the estimator accepts any probabilistic source classifier. For bounded additive losses and i.i.d. target samples, a uniform bound, which we use as a diagnostic, separates the target sample size from the mismatch between the proxy and the deployment model. A second result gives the error floor of symmetric scores under asymmetric transfer.

- **An evaluation that separates what each method may read.** We group methods by whether they need a trained model per source or the transfer matrix, and count separately the one-off matrix that sets our three hyperparameters. We split by city, repeat each transfer over seeds, score two deployment models separately and jointly, and confirm on the Sionna test cities.

With 1000 labeled samples of each Sionna target, enough to measure ranking quality precisely, DiMTE ranks sources at 0.78 and 0.79, better than every method that does not train a deployment model per source, the strongest of which reaches 0.75 and 0.68 (Table 1). On the DeepMIMO test cities the loss of its top-ranked source only ties the strongest comparator's, and NCE on a CNN trained for every source ranks Sionna LOS/NLOS sources higher. The gain lies in comparing a whole library at low cost, not in picking a single winner.

2 Related Work

Transferability estimators rank candidate pretrained models for a labeled target task. Prediction-based criteria such as LEEP, NCE and N-LEEP read a source model's outputs on target examples (Nguyen et al., 2020; Tran et al., 2019; Li et al., 2021), while feature-based criteria measure target separability or representation geometry (You et al., 2021; Bao et al., 2019; Ibrahim et al., 2022; Pándy et al., 2022; Huang et al., 2022; Bolya et al., 2021; Tan et al., 2021; Shao et al., 2022; Wang et al., 2023; Gholami et al., 2023; Zhang et al., 2025b). All address model selection, where each candidate carries a trained model, while we address the earlier choice of source *dataset* and include them as a richer regime.

Dataset-level methods compare empirical distributions or embed each dataset into a vector, without a deployment model per source: OTDD transports labeled examples across datasets (Alvarez-Melis & Fusi, 2020), Cui et al. (2018) select pretraining data by the earth mover's distance in one pretrained feature space, Task2Vec, Dataset2Vec and Domain2Vec learn task or dataset representations (Achille et al., 2019; Jomaa et al., 2021; Zhang et al., 2025a), and wireless work compares class-aware centroids and sliced-Wasserstein distances (Morais et al., 2026). All quantify similarity, while transfer concerns the risk of a model trained on one dataset and evaluated on another, which is directed. Directed KL comparisons (Westny et al., 2026), box embeddings that order tasks rather than metrize them (Daroya et al., 2024), label-conditional transport (Tan et al., 2021; 2024) and $\mathcal{H}$-divergence and Wasserstein bounds on the target risk (Ben-David et al., 2010; Redko et al., 2017; Zhan et al., 2026) each reach for that asymmetry, yet none returns a value in the deployment loss. We run OTCE, JC-NCE, WDJE, and LEEP and NCE on our proxy in our encoder space, so none of them needs a trained source extractor. Our Deep Sets and Set Transformer baselines learn directed pairwise outcomes (Zaheer et al., 2017; Lee et al., 2019), at the cost of needing a transfer matrix to train.

Frozen-feature nearest-neighbor and linear probes have served as cheap proxies for expert and model selection (Puigcerver et al., 2021; Renggli et al., 2022), but they read a trained model per candidate, as do Taskonomy (Zamir et al., 2018), DsDm (Engstrom et al., 2024) and the symmetric cross-learning score (Sun et al., 2025), which measure or predict transfer from models trained on candidate data. Label-free accuracy estimators (Guillory et al., 2021; Garg et al., 2022) and data attribution by influence functions or datamodels (Koh & Liang, 2017; Ilyas et al., 2022) likewise start from a trained model. Our default proxy is mixture discriminant analysis (Hastie & Tibshirani, 1996), whose components act as class prototypes (Snell et al., 2017). Our contribution is to evaluate a source-induced proxy on target data as a directed estimate of a *different, specified model's* transfer risk, in the deployment loss, and to test when it succeeds. Table 9 places every family on four axes.

3 Problem Formulation

Domains, training procedure, and transfer. Let D_i be a labeled sample from domain P_i over inputs x, labels y in a global vocabulary $\mathcal{Y}$ that a domain may use only in part, and optional evaluation metadata a such as a target beam-power vector. The scope is common-vocabulary classification, not regression or a mismatched label space. Let $\mathcal{A}$ be a specified deployment training procedure, with its architecture, optimization and source-data budget, let ξ collect its training randomness, and write $h_{s,\xi} = \mathcal{A}(D_s^{\mathrm{tr}}; \xi)$ for the deployment model it trains on source s. For a lower-is-better target evaluation functional $\mathcal{M}_t$, define

$$T_{\mathcal{A}}(s \to t) = \mathbb{E}_{\xi}\, \mathcal{M}_t(h_{s,\xi}). \tag{1}$$

For an additive task loss, $\mathcal{M}_t(h) = \mathbb{E}_{(x,y,a)\sim P_t}\ell(h(x), y, a)$. For LOS/NLOS identification, $\mathcal{M}_t(h) = 1 - F_{1,\mathrm{w}}(C_t(h))$, where $[C_t(h)]_{yc} = \Pr_t[Y = y,\, h(X) = c]$ is the population confusion matrix. Transfer therefore depends on the task, the data budget and the training procedure. Experiments estimate (1) on a target test split, disjoint from the training split of the same seed, and average over training seeds (Section 5.2). We write P_{st} for this measured transfer. Our main results average it over two deployment models that share no representation, their *two-learner mean*, so that no single architecture determines it, and we also report each model alone.

For beam prediction, $\mathcal{M}_t$ is the mean Layer-1 reference-signal received-power (L1-RSRP) difference in dB between the predicted and the genie-aided beam, that is, the power a user loses when served by the predicted beam instead of the best one. A beam's L1-RSRP is the power of its reference signal that a user measures in a sweep, so the loss needs the beam-power vector a of each labeled sample. We also report Top-1/1 accuracy, the share of users whose best beam is predicted. Section 4.3 covers bounded additive losses, and weighted F1 uses a soft-confusion plug-in with the fixed-sample lemma of Appendix A. The plug-in is biased, yet reading out accuracy ranks sources alike.

Access regime. For a new target, we assume access to (i) labeled source datasets or their stored summaries, (ii) a shared frozen encoder, and (iii) a labeled target sample V_t of m examples. The headline experiments read $m = 1000$ to measure ranking quality precisely. A new site rarely has that many labels, so Section 6.2 lowers m to two on LOS/NLOS identification and to ten on beam prediction. We do *not* assume a trained deployment model for every source, which differs from model-based transferability estimation and is stronger than a label-free dataset comparison. The pseudo-label procedures of Appendix L are task-specific and not part of the generic estimator.

Selection and evaluation. For a candidate set $\mathcal{S}_t$, the genie-aided (best in hindsight) and estimated selections are

$$s^*(t) = \arg\min_{s\in\mathcal{S}_t} T_{\mathcal{A}}(s\to t), \qquad \hat{s}(t) = \arg\min_{s\in\mathcal{S}_t} \widehat{P}_{s\to t}. \tag{2}$$

The object of interest is the full vector $T_{\mathcal{A}}(\mathcal{S}_t \to t)$, which we score by mean absolute error in task units and by ρ, the Spearman correlation between estimated and measured transfer over the candidates of one target, averaged over targets. We also report the regret of the selected source, $T_{\mathcal{A}}(\hat{s}\to t) - T_{\mathcal{A}}(s^*\to t)$, and the gap it recovers, the share of the loss difference between a random and the genie-aided source, pooled over targets, because a reliable ranking need not improve the single top choice when several sources nearly tie.

4 Source-Induced Proxy Risk

4.1 Generic Estimator

Direct evaluation of $T_{\mathcal{A}}(s\to t)$ requires training h_s. DiMTE replaces it with a lightweight classifier g_s built from source D_s, the proxy. Let ϕ be a frozen encoder and let a fixed preprocessing map W produce $u = W(\phi(x)) \in \mathbb{R}^k$. Given a probabilistic proxy $p_s(c \mid u)$ and a labeled target sample $V_t = \{(x_i, y_i, a_i)\}_{i=1}^m$, define

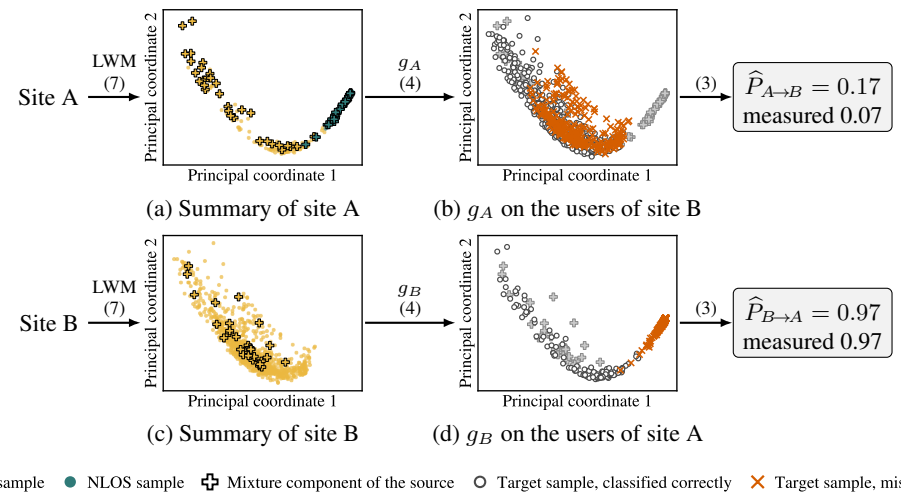
$$\widehat{P}_{s\to t} = \frac{1}{m}\sum_{i=1}^{m}\sum_{c\in\mathcal{Y}_s} p_s(c \mid u_i)\,\ell(c, y_i, a_i). \tag{3}$$

The source enters through p_s and the target through the empirical expectation, so generally $\widehat{P}_{s\to t} \neq \widehat{P}_{t\to s}$. The score estimates a target loss, not a distance, so it need not be symmetric, vanish on the diagonal or satisfy a triangle inequality.

Weighted F1 is not additive, so we form the soft confusion matrix $\widehat{C}_{yc} = \sum_{i:y_i=y} p_s(c \mid u_i)$ and plug it into the standard weighted-F1 functional, so that $\widehat{P}_{s\to t} = 1 - F_{1,\mathrm{w}}(\widehat{C})$, a deterministic surrogate rather than the expected F1 of a randomized classifier.

4.2 A Fixed-Size Source Proxy

In the wireless instantiation ϕ is the frozen encoder of the Large Wireless Model (LWM), a wireless foundation model (Alikhani et al., 2024; 2025), and $z = \phi(x)$ is its ℓ_2-normalized `[CLS]` representation, which the preprocessing map centers, projects onto k principal directions and divides by



(a) Summary of site A (b) g_A on the users of site B

(c) Summary of site B (d) g_B on the users of site A

Figure 2: How DiMTE computes the two directions of Figure 1. The panels draw the first two of the k standardized coordinates of Section 4.2, and decisions read all k. (a) Site A keeps 32 components per class, and NLOS users make up 88% of its samples. (b) The classifier g_A labels 72% of site B's users correctly, so the estimate is low. (c) Site B has no NLOS users, so its summary is LOS only. (d) Its classifier g_B therefore calls every user of site A LOS and misclassifies the NLOS users, 88% of the site, so the estimate is high.

the fitted standard deviation of each coordinate. A shared map puts every source and target in one coordinate system. Its k decorrelated directions suit a diagonal variance fitted from a few hundred samples per class, and standardizing keeps the leading direction from dominating the distances (Table 15). The tables fit this map on the unlabeled embeddings of the collection they score, and Table 1 also reports a map fitted on the training cities alone, so that no test sample shapes the space (Appendix F). The bound below covers that second map.

For each active source class $c \in \mathcal{Y}_s$, k-means produces $M_{sc} \le \min(M, n_{sc})$ prototypes μ_{scj} with weights $w_{scj} = n_{scj}/n_{sc}$, class priors π_{sc} (add-one frequencies), and a single diagonal variance Σ_s that every class shares, shrunk toward the source's pooled variance and tied by class support (Appendix A.1; Figure 2 draws two summaries). Several prototypes per class let one class span several propagation regions. The tied variance removes class-specific quadratic terms from the decision boundaries, which a few hundred samples per class cannot estimate reliably, and it matters most among the summary's choices (Tables 15 and 16). The proxy logits and probabilities are

$$a_{sc}(u) = \beta \log \pi_{sc} + \log \sum_{j=1}^{M_{sc}} w_{scj} \exp\left[-\tfrac{1}{2}(u-\mu_{scj})^\top \Sigma_s^{-1}(u-\mu_{scj})\right], \qquad p_s(c \mid u) = \mathrm{softmax}_{c\in\mathcal{Y}_s}(a_{sc}(u)/\tau). \tag{4}$$

We set the prior weight $\beta = 1$, and the softmax temperature τ gives hard decisions as $\tau \to 0$. A class absent from the source has zero predictive mass. A model trained on that source rarely predicts such a class either, as for B→A in Figure 2d, estimated at 0.97 and measured at 0.97. The summary stores at most $CM(k+1)+k+C$ scalars and scoring one source costs $O(mCMk)$, so M trades storage for flexibility. One prototype per class resembles linear discriminant analysis, and one per sample approaches the Gaussian-kernel classifier that the k-NN and Parzen variants store (Appendix K).

4.3 What the Proxy Approximation Requires

The central modeling substitution in DiMTE is g_s for the deployment model. For a probabilistic proxy, let $\bar{\ell}_s(x,y,a) = \sum_c p_s(c \mid u)\ell(c,y,a)$ denote its conditional expected task loss. The following result makes the consequence explicit. It is a diagnostic rather than a certificate, because its mismatch term depends on the trained deployment models, and Table 4 measures both of its terms.

Proposition 1 (Uniform proxy-risk bound)**.** *Condition on fixed source samples and proxies. Fix a target-independent representation and preprocessing map, let $N = |\mathcal{S}_t|$, and assume an additive loss $\ell \in [0, B]$. Let*

$$\Delta_{s,t} = \mathbb{E}_{\xi,(x,y,a)\sim P_t}\left|\bar{\ell}_s(x,y,a) - \ell(h_{s,\xi}(x),y,a)\right|.$$

For an i.i.d. labeled target sample of size m, with probability at least $1-\delta$, simultaneously for every candidate source,

$$\left|\widehat{P}_{s\to t} - T_{\mathcal{A}}(s\to t)\right| \le \Delta_{s,t} + B\sqrt{\frac{\log(2N/\delta)}{2m}}. \tag{5}$$

Two triangle inequalities then bound the regret of the selected source by $\Delta_{\hat{s},t} + \Delta_{s^*,t}$ plus twice the concentration term (Appendix A).

The proof is a uniform Hoeffding bound followed by two triangle inequalities (Appendix A), and for hard predictions under a bounded class-dependent loss $\Delta_{s,t} \le B\,\mathbb{E}_\xi \Pr_t[g_s(x) \ne h_{s,\xi}(x)]$. It assumes true target labels, so it covers neither the pseudo-labels of Section 6.2 nor the correlation of neighboring users on a ray-traced grid. On LOS/NLOS identification, with the classification error as the loss ($B = 1$), the proxy's hard decisions and a CNN trained on raw channels disagree on 7.9% of the Sionna samples, and the bound reads 0.142 (mismatch 0.079, sampling 0.064) where the error is 0.044 (Table 4). The L1-RSRP difference has no fixed bound, and with each target's largest loss as B the bound averages 62.3 dB against an error of 4.4 dB, too loose to order sources. We therefore read Proposition 1 as a diagnostic, and the ranking claims rest on the measurements. Two signals that need no deployment model flag the largest mismatches on LOS/NLOS identification (Appendix A.2).

A symmetric score carries a separate cost, since its mean absolute error is at least half the average directional discrepancy (Proposition 3). In our matrices that asymmetry is 2.2 to 5.0 times the variation between training seeds (Appendix B). The floor says no symmetric score reproduces an asymmetric transfer matrix, not that DiMTE ranks sources well, which we test directly.

5 Experimental Protocol

This section defines the data, the measured transfer, what each method may read, and the tests.

5.1 Wireless Tasks and Splits

Each dataset is one site in one ray-traced city, and a sample is the narrowband channel from its 32 antennas to one user, the input format of LWM (Appendix C). LOS/NLOS identification keeps users with at least one path. Beam prediction reads the channel state information (CSI) of an eight-antenna sub-array and predicts the best of the full array's 32 fixed beams.

Cities, not sites, define the split. The 44 training cities, with 68 datasets, choose k, M and τ and train the set encoders. LWM was pretrained without labels on channels of 20 of these cities and of no test city (Appendix C). The 59 DeepMIMO test cities were scored during design, so they corroborate rather than confirm. The 84 datasets of the 42 Sionna test cities were ray traced after the estimator and its settings were fixed (Appendix D), so they give the confirmatory results of Section 6.1.

A target's candidate sources are the datasets of the other cities of its own test set.

5.2 Measured Transfer and Access Regimes

We measure transfer for four deployment models whose recipes come from within-source validation: (i) a linear classifier and (ii) an MLP on frozen LWM features, (iii) fine-tuned LWM, and (iv) a CNN trained from scratch on raw CSI (Appendix E). The two-learner mean of Section 3 uses the last two. On the training cities their source rankings agree at $\rho = 0.84$ and 0.74, while the MLP differs (Figure 7). Each matrix averages two or three training seeds (Appendix B).

Baselines are grouped by what they may read (Table 8): (i) training-free scores read source and target samples but no trained deployment model, (ii) set encoders also read the transfer matrix of the training cities, and (iii) model-aware estimators receive a trained model for every source. DiMTE needs no source deployment model to score a target, but its k, M and τ come from that training-city matrix, measured once per task with about 1.4 and 3.6 GPU-hours of training (Table 11). A 1-NN rule that reuses the LOS/NLOS $k = 32$ needs no new matrix and, in a later check, stays within 0.03 of the tuned estimate on every set (Table 13; Figure 6 varies each setting).

Every comparison is paired over the targets and clustered by city, since the sites of one city share buildings and propagation. The one-sided p-value comes from a sign-flip permutation test that

exchanges the two methods for all base stations of a city at once. The interval is a percentile bootstrap over the same clusters (Appendix K). Neither covers the training-seed noise of the measured transfer. Redrawing those seeds 500 times gives our correlation a standard deviation of at most 0.011 and never reverses our lead over a comparator (Appendix B).

6 RESULTS

Number pairs give LOS/NLOS identification first, and ρ is the mean per-target Spearman correlation.

6.1 HELD-OUT TEST: TRANSFER PROFILES ON NEW CITIES

On the Sionna test cities DiMTE ranks sources better than all 48 LOS/NLOS and 54 beam comparators of Table 23. Each margin over the strongest comparator has a one-sided p below 0.001 (Table 1). The beam lead is wider because DiMTE reads out the L1-RSRP difference from the labeled users' beam powers, and comparators that read the same powers reach at most 0.67. From beam labels alone, DiMTE still leads every comparator on the DeepMIMO test cities (0.81) but reaches 0.65 on the Sionna test cities, below directed joint KL on raw channels at 0.68 (Table 34).

Table 1: Source ranking against the measured transfer, the mean of fine-tuned LWM and a raw-CSI CNN. The Sionna test cities were ray traced after the design was frozen, and the DeepMIMO test cities were read during design. Only the latter have labels for a third task, uplink angle-of-arrival (UL-AoA) sectors (Appendix Q). Each baseline row is the strongest of its family by mean ρ, and *strongest comparator* is the best method of each column among all methods of Tables 20 and 23. ρ is the per-target Spearman correlation, with the standard error of its city means in the last place. *Gap* is the share of the loss difference between a random and the genie-aided source that the selected source recovers, and the last column of a task is the loss that source delivers. Best in bold and second best underlined, excluding the estimators' row. Shaded rows use the paper's setting.

	Sionna test cities						DeepMIMO test cities							
	LOS/NLOS			Beam			LOS/NLOS			Beam			UL-AoA	
Score	ρ	gap	1−F1	ρ	gap	dB	ρ	gap	1−F1	ρ	gap	dB	ρ	gap
Training-free scores														
Class-prior KL	0.16(4)	33	0.073	0.65(1)	64	10.3	0.36(2)	33	0.131	0.73(2)	70	6.4	0.66(1)	60
OTDD on raw CSI	0.34(1)	57	0.052	0.67(1)	72	9.7	0.22(1)	29	0.138	0.72(1)	65	6.7	0.61(1)	59
Joint KL on LWM	0.40(3)	75	0.036	0.62(2)	74	9.5	0.48(2)	80	0.063	0.77(2)	78	6.0	0.68(1)	77
NCE on our mixture	0.73(2)	87	0.026	0.57(2)	40	12.4	0.72(2)	78	0.066	0.74(3)	63	6.9	0.79(2)	80
Class-prior KL on L1-RSRP	–	–	–	0.67(1)	65	10.3	–	–	–	0.74(2)	70	6.5	–	–
Learned set encoders														
Deep Sets, self-supervised	0.38(3)	66	0.044	0.37(2)	53	11.3	0.27(2)	61	0.090	0.39(1)	67	6.6	0.27(2)	65
Deep Sets on the transfer matrix	0.58(3)	61	0.049	0.44(1)	38	12.5	0.51(4)	72	0.074	0.69(2)	61	6.9	0.72(1)	71
DiMTE with each source classifier (ours)														
Mixture summary (default)	**0.78**(1)	**88**	**0.025**	0.79(1)	74	9.5	0.85(1)	85	0.055	**0.91**(1)	79	5.9	**0.94**(1)	**92**
k-NN rule	0.76(1)	**88**	**0.025**	0.79(1)	**81**	**8.9**	**0.87**(1)	**89**	**0.050**	**0.91**(1)	81	5.8	0.93(1)	91
Parzen rule	0.76(1)	**88**	**0.025**	**0.82**(1)	80	9.0	**0.87**(1)	**89**	**0.050**	0.90(1)	**83**	**5.7**	**0.94**(1)	91
projection from training cities only	0.76(1)	87	**0.025**	0.79(1)	74	9.5	0.84(1)	83	0.058	**0.91**(1)	79	5.9	–	–
Strongest comparator (all methods)	0.75(2)	87	0.026	0.68(1)	67	10.2	0.72(2)	82	0.060	0.77(2)	78	6.0	0.79(2)	80
Estimators with a model per source, best of 24	0.85	84	0.028	0.70	73	9.6	0.76	89	0.050	0.78	72	6.3	–	–
Random source	–	0	0.102	–	0	15.7	–	0	0.180	–	0	10.6	–	0
Genie-aided source	–	100	0.014	–	100	7.4	–	100	0.034	–	100	4.6	–	100

The lead does not come from fitting the projection on the test samples. With directions from the training cities, DiMTE still reaches 0.76 and 0.79, above every comparator that also fits on the training cities (each p below 0.001; Table 1). Two directed scores in the same space, the transferability estimator JC-NCE and the risk bound WDJE, reach at most 0.19 and 0.43 (Appendix E). Against each deployment model alone, DiMTE leads on both tasks against the CNN, and against fine-tuned LWM, whose backbone it shares, only on beam prediction (0.64 against 0.66 on LOS/NLOS; Table 24).

A library built with one ray tracer also ranks sources for targets built with another. Because an operator holds a library built earlier, we also rank the 152 DeepMIMO datasets for the 84 Sionna targets, against one raw-CSI CNN per source (Appendix M). DiMTE leads all 12 comparators of that study on beam prediction (all p below 0.001), while on LOS/NLOS identification its lead over LEEP on our mixture is within the noise ($p = 0.056$; Table 2). On beam prediction its top-ranked source recovers 45% of the gap to the genie-aided source, less than our nearest-class-mean variant (65%).

Table 2: Source ranking in four further settings: (i) the 152 DeepMIMO datasets for Sionna targets, against one raw-CSI CNN per source, (ii) Sionna targets with ten labeled samples, over twenty draws, (iii) camera-trap locations, and (iv) MASSIVE source languages. The rank correlation ρ and the *gap* are as in Table 1, with the standard error over clusters. *Best other* is the strongest method that trains no model per source and reads the same labels, and p the largest one-sided value over them.

		Rank correlation ρ			Gap recovered		
Decision	Setting	DiMTE	Best other	Δ	DiMTE	Best other	p
Rank across simulators	LOS/NLOS, 84 targets	0.75(1)	0.73	+0.02	87%	87%	0.056
	Beam, 84 targets	0.75(1)	0.55	+0.20	45%	38%	<0.001
Rank from ten labels	LOS/NLOS, 84 targets	0.55	0.54	+0.01	61%	62%	0.120
	Beam, 84 targets	0.62	0.57	+0.05	63%	56%	<0.001
Rank camera traps	DINOv2, 29 targets	0.87(2)	0.81	+0.06	86%	82%	<0.001
Rank source languages	mBERT, 27 targets	0.66(2)	0.67	−0.01	66%	75%	0.820

Estimators with a model per source win one of the four comparisons of Table 1, Sionna LOS/NLOS identification (NCE on the CNN, 0.85). Self-agreement explains part of that win, since the CNN is one of the two models the measured transfer averages. Scored against the deployment model that each method does not read, DiMTE leads both estimators on both tasks (Tables 27 and 28). Their remaining lead on the two-learner mean costs a trained model per source (Figure 1).

On the DeepMIMO test cities the source DiMTE selects only ties the strongest comparator's in regret (Appendix K), so the gain lies in the order of the whole library. In task units the raw estimate is pessimistic, because the proxy is weaker than a trained model. On the Sionna test cities it errs by 0.049 in $1-$F1 and 4.4 dB, against 0.088 and 4.8 dB for a constant (Appendix K.1). An exploratory isotonic map fitted on training-city pairs brings it to 0.038 and 2.3 dB, against 0.054 and 3.1 dB for the best other score so mapped (Table 39). The map is monotone, so it changes no ranking, gap or regret. On beam prediction, an offset from one trained model already gives 2.5 dB (Appendix K.1).

Table 3: What the selected data deliver on Sionna test sites, in an exploratory study designed after the first Sionna results. A site with a hundred labeled samples either adds datasets of the DeepMIMO library to them (first block) or trains on library datasets alone (second block). Each entry trains the raw-CSI CNN once per site and reports the error on the site's held-out users, in $1-$F1 and in L1-RSRP difference (dB), lower is better. *Best other score* is the strongest rule given the same labels, LEEP on our mixture for LOS/NLOS identification and the class-prior KL for beam prediction. Sites are the first Sionna datasets in alphabetical order, 20 and 16 for the first block and 20 and 20 for the second. The standard error over cities is in the last place. Best of each block in bold, and shaded rows use the paper's setting.

Decision and what the site trains on	LOS/NLOS	Beam prediction
What to add to the site's own 100 *labels*		
The site's samples alone	**0.012**(2)	7.54(60)
+ the one DiMTE ranks first	0.018(3)	6.63(43)
+ the best other score's first	0.019(3)	7.38(84)
+ the three DiMTE ranks first	0.021(4)	**5.86**(56)
+ the best other score's first three	0.024(5)	8.19(112)
+ three random datasets	0.037(4)	10.93(101)
What to train on when the site trains on none of its own		
The three DiMTE ranks first	**0.032**(7)	**8.74**(83)
The best other score's first three	0.034(7)	12.00(112)
Three random datasets	0.089(23)	14.92(105)
The dataset DiMTE ranks last	0.823(25)	27.07(90)

6.2 A Site With Few Labels

With ten labeled samples and nothing else, the unchanged estimator ranks sources at 0.55 and 0.62 (Table 2). On beam prediction it leads every rule given the same labels, by +0.05 at ten labels and at every larger budget (largest p below 0.001; Table 41). On LOS/NLOS identification its lead over the closest rule is within the noise (largest p 0.120).

In an exploratory correction, the site's unlabeled channels also help. A training-city classifier pseudo-labels them for LOS/NLOS identification, and a sub-array sweep does so for beam prediction

(Appendix L, also with noisy sweeps). The ten labeled samples then correct the pseudo-labeled estimate (Angelopoulos et al., 2023), which raises ρ to 0.66 and 0.71. The same samples lift LEEP and NCE on our mixture to 0.65 on LOS/NLOS identification, level with ours, and the class-prior KL to 0.64 on beam prediction (Table 44).

On beam prediction with a hundred labeled samples, the datasets DiMTE ranks first improve a site's model, and alone they train a better one than the comparator's choices. Ranking quality matters through the model a site trains, so an exploratory study trains the raw-CSI CNN on data selected from the DeepMIMO library and tests it on held-out users of Sionna sites (Table 3). The comparator is the strongest other rule given the same labels. On beam prediction the three datasets DiMTE ranks first lower the site's L1-RSRP difference from 7.54 to 5.86 dB when added ($p < 0.01$), and alone they train a better model than the comparator's three (p 0.022 over 10 cities). As additions, the two selections do not differ significantly (p 0.052 over 8 cities). On LOS/NLOS identification a hundred labels nearly solve the task (0.012 in $1-$F1), every addition costs a little and the two rules choose equally well, yet the dataset DiMTE ranks last trains a model at 0.823 in $1-$F1.

Direction carries much of the ranking quality. A symmetrized DiMTE ranks Sionna sources at 0.55 instead of 0.78 on LOS/NLOS identification (Table 25), and the asymmetry it predicts tracks the measured one (Figure 13), so a symmetric score discards signal rather than noise. The ranking is also specific to the target, not a list of generally good sources. DiMTE still ranks Sionna sources at 0.76 and 0.80 when each source's mean transfer to the other targets is removed from the estimate and from the measurement, while the best comparator falls to 0.59 and 0.66 (Table 35). The source classifier matters much less, since the mixture, k-NN and Parzen rules rank within a few hundredths while one mean per class and a trained linear probe fall clearly behind (Table 29).

The representation decides where proxy risk tracks the deployment model. Beam prediction does not need the LWM encoder. On raw channels that keep the received power, the estimator ranks Sionna beam sources at 0.80 with the k-NN rule and 0.76 with the mixture, above the strongest comparator (Table 30). On those channels LOS/NLOS identification reaches at most 0.72, against 0.78 on LWM embeddings, and 0.21 once each channel is normalized as the input of LWM is. On the frozen features of WavesFM (Aboulfotouh & Abou-Zeid, 2025), a public foundation model that shares no weights with LWM or the deployment models, the estimate also leads every comparator with the same access on the Sionna cities, and ties the best on DeepMIMO beam prediction (Appendix I).

6.3 Cost, Storage, and Other Domains

Once the samples are embedded, DiMTE reads one target against all 236 datasets in 0.25 to 3.44 CPU seconds. A model-aware estimator first trains a deployment model for each of the 236 sources, 0.29 to 5.6 GPU-hours, for every task and deployment model (Table 11). The mixture keeps 64 vectors per source on LOS/NLOS identification, 15.7 times fewer than the k-NN and Parzen proxies store, but 940 of 1000 at the beam setting (Figure 15), so the library stores data or summaries, not models.

Outside wireless the estimate needs only a frozen encoder, class labels and a task metric (Table 2). On camera traps (Beery et al., 2020) it leads clearly (Appendix O). On source-language selection for MASSIVE, a multilingual intent benchmark (FitzGerald et al., 2023), it ties the best comparator, because the source language decides most of the transfer (Table 47 and Appendix P).

7 Limitations and Conclusion

The evidence has four limits. First, the estimator reads a labeled or pseudo-labeled target sample, its pseudo-labeler is task-specific, and the beam loss needs each labeled user's beam-power vector. Second, every task we study is classification at equal source-data budgets, and Appendix K.2 relaxes the shared label set only for coarser beam labels. Third, one proxy serves every deployment model, so the estimate tracks some better than others (Table 24), and it needs a representation that keeps the class structure they use (Section 6.2). Fourth, every channel is ray traced at one carrier frequency and with one antenna array, although by two different ray tracers. Within these limits, DiMTE keeps most of the ranking quality of training a deployment model on every source, at a fraction of its compute (Figure 1). What changes is the price of the decision. Once a task's hyperparameters are fixed, a practitioner ranks hundreds of datasets in the deployment metric in CPU seconds, before training any model on them.

Ethics statement

The wireless experiments use ray-traced channels of simulated users and contain no personal data. The Sionna test cities are ray traced on building footprints from OpenStreetMap (© OpenStreetMap contributors, Open Database License). The camera-trap images come from iWildCam2020-WILDS (CDLA-Permissive 1.0) and the utterances from MASSIVE (CC BY 4.0), and every dataset is used under its license. A poor source recommendation can lower the performance of the deployed model. DiMTE should therefore supplement, not replace, validation at the target site.

Reproducibility statement

Section 3 defines the estimand, Section 4 the estimator and Section 5 the protocol. The appendices document the data, deployment-model recipes, baseline access regimes, hyperparameters, compute, transfer-matrix reliability and negative results. Every reported result in the text and the tables is generated by script from the stored result files. We will release the code with the final paper, together with the DeepMIMO scenario lists, the centers of the Sionna test cities, the script that ray traces them, the measured transfer matrices and the result files.

AI use statement

Generative AI tools were used under the authors' direction. They wrote and ran parts of the experiment and analysis code and generated tables and figures from the result files. They proposed experiments and controls, some of which the authors adopted, and helped find, verify and format references. They also helped draft and edit the text. The authors formulated the research questions and the method, made every scientific decision, checked every reported result and take full responsibility for the manuscript.

References

3GPP. TR 38.843: Study on artificial intelligence (AI)/machine learning (ML) for NR air interface. Technical Report 3GPP TR 38.843 V18.0.0, 3rd Generation Partnership Project (3GPP), December 2023. Release 18.

Ahmed Aboulfotouh and Hatem Abou-Zeid. Multimodal wireless foundation models. *arXiv preprint arXiv:2511.15162*, 2025.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6430–6439, 2019.

Fayçal Aït Aoudia, Jakob Hoydis, Merlin Nimier-David, Baptiste Nicolet, Sebastian Cammerer, and Alexander Keller. Sionna RT: Technical report. *arXiv preprint arXiv:2504.21719*, 2025.

Sadjad Alikhani, Gouranga Charan, and Ahmed Alkhateeb. Large wireless model (LWM): A foundation model for wireless channels. *arXiv preprint arXiv:2411.08872*, 2024.

Sadjad Alikhani, Gouranga Charan, and Ahmed Alkhateeb. LWM: A pre-trained wireless foundation model for universal feature extraction. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pp. 1–6, 2025. doi: 10.1109/ICMLCN64995.2025.11140266.

Ahmed Alkhateeb. DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications. In *Information Theory and Applications Workshop (ITA)*, 2019.

David Alvarez-Melis and Nicolò Fusi. Geometric dataset distances via optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

Sara Beery, Elijah Cole, and Arvi Gjoka. The iWildCam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175, 2010.

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8440–8451, 2020.

Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Rangel Daroya, Aaron Sun, and Subhransu Maji. Task2Box: Box embeddings for modeling asymmetric task relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. DsDm: Model-aware dataset selection with datamodels. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 12491–12526, 2024.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4277–4302, 2023.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations (ICLR)*, 2022.

Mohsen Gholami, Mohammad Akbari, Xinglu Wang, Behnam Kamranian, and Yong Zhang. ETran: Energy-based transferability estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Berkay Guler, Giovanni Geraci, and Hamid Jafarkhani. A multi-task foundation model for wireless channel representation using contrastive and masked autoencoder learning. *IEEE Journal on Selected Areas in Communications*, 44:4489–4504, 2026. doi: 10.1109/JSAC.2026.3677157.

Trevor Hastie and Robert Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):155–176, 1996.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Long-Kai Huang, Junzhou Huang, Yu Rong, Qiang Yang, and Ying Wei. Frustratingly easy transferability estimation. In *International Conference on Machine Learning (ICML)*, 2022.

Shibal Ibrahim, Natalia Ponomareva, and Rahul Mazumder. Newer is not always better: Rethinking transferability metrics, their peculiarities, stability and performance. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2022)*, Lecture Notes in Computer Science, pp. 693–709. Springer, 2022. doi: 10.1007/978-3-031-26387-3_42.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587, 2022.

Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. Dataset2vec: Learning dataset meta-features. *Data Mining and Knowledge Discovery*, 35:964–985, 2021. arXiv:1905.11063.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 5637–5664, 2021.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3125–3135, 2019.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Volume 2, Short Papers*, pp. 8–14, 2017.

Boxun Liu, Shijian Gao, Xuanyu Liu, Xiang Cheng, and Liuqing Yang. WiFo: Wireless foundation model for channel prediction. *Science China Information Sciences*, 68(6):162302, 2025. doi: 10.1007/s11432-025-4349-0.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2020.

João Morais, Sadjad Alikhani, Akshay Malhotra, Shahab Hamidi-Rad, and Ahmed Alkhateeb. Wireless dataset similarity: Measuring distances in supervised and unsupervised machine learning. arXiv preprint arXiv:2601.01023, 2026.

Cong T. Nguyen, Nguyen Van Huynh, Nam H. Chu, Yuris Mulya Saputra, Dinh Thai Hoang, Diep N. Nguyen, Quoc-Viet Pham, Dusit Niyato, Eryk Dutkiewicz, and Won-Joo Hwang. Transfer learning for wireless networks: A comprehensive survey. *Proceedings of the IEEE*, 110(8):1073–1115, 2022. doi: 10.1109/JPROC.2022.3175942.

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cédric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning (ICML)*, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. Transferability estimation using bhattacharyya class separability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cedric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. Scalable transfer learning with expert models. In *International Conference on Learning Representations*, 2021.

Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, Lecture Notes in Computer Science, pp. 737–753. Springer, 2017. doi: 10.1007/978-3-319-71246-8_45.

Cedric Renggli, André Susano Pinto, Luka Rimanic, Joan Puigcerver, Carlos Riquelme, Ce Zhang, and Mario Lučić. Which model to transfer? Finding the needle in the growing haystack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging Fisher space. In *European Conference on Computer Vision*, 2022.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Shudong Sun, Hao Helen Zhang, and Joseph C. Watkins. Quantifying data similarity using cross learning. *arXiv preprint arXiv:2510.10866*, 2025.

Yang Tan, Yang Li, and Shao-Lun Huang. OTCE: A transferability metric for cross-domain cross-task representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yang Tan, Yang Li, and Shao-Lun Huang. Practical transferability estimation for image classification tasks. *arXiv preprint arXiv:2106.10479*, 2024.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Anh T. Tran, Cuong V. Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual E5 text embeddings: A technical report, 2024.

Zijian Wang, Yadan Luo, Liang Zheng, Zi Huang, and Mahsa Baktashmotlagh. How far pre-trained models are from neural collapse on the target dataset informs their transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Theodor Westny, David Axelsson, Björn Olofsson, and Erik Frisk. Unveiling transferability in trajectory prediction via latent scene embeddings. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 41–63. Springer, 2026. doi: 10.1007/978-3-032-37461-5_3.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning (ICML)*, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3712–3722, 2018.

Qianshan Zhan, Xiao-Jun Zeng, and Qian Wang. To transfer or not to transfer: Unified transferability metric and analysis. *IEEE Transactions on Cybernetics*, 56(6):3551–3564, 2026. doi: 10.1109/TCYB.2026.3656684.

Mozhi Zhang, Howe Tissue, Lu Wang, and Xipeng Qiu. Domain2vec: Vectorizing datasets to find the optimal data mixture without training. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 76139–76158, 2025a. arXiv:2506.10952.

Tengxue Zhang, Yang Shu, Xinyang Chen, Yifei Long, Chenjuan Guo, and Bin Yang. Assessing pre-trained models for transfer learning through distribution of spectral components. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22560–22568, 2025b. doi: 10.1609/aaai.v39i21.34414.

GUIDE TO THE APPENDICES

The appendices fall into six groups: (i) the formal results and the noise of the measured transfer (Appendices A and B), (ii) the wireless data (Appendices C and D), (iii) the implementation, the hyperparameters, the ablations and other encoders (Appendices E to I), (iv) the detailed wireless results (Appendices J to N), (v) camera traps, source languages and a third wireless task (Appendices O to Q), and (vi) the history of the protocol and the terms (Appendices R and S). The table below gives, for each appendix, the question it answers and the main-text claim it supports.

The question that each appendix answers and the main-text claim it supports.

App.	Question it answers	Main-text claim it supports
A	When does the proxy track a deployment model?	Bound, F1 lemma, symmetric floor (§3, §4.3)
B	How noisy and how directed is measured transfer?	Asymmetry beyond seed noise (§4.3, §5.2)
C	What does a wireless dataset contain?	Tasks, samples and splits (§5.1)
D	How were the Sionna test cities built?	Confirmatory test (§5.1, §6.1)
E	How are models, baselines and costs set up?	Matched access and cost (§5.2, §6.3)
F	How were k and τ chosen?	Hyperparameters from training cities (§4.2, §5.2)
G	How many components per class?	Choice of M (§4.2)
H	Which choices of the summary matter?	Projection and tied variance (§4.2)
I	Does DiMTE need the LWM encoder?	Lead on WavesFM features (§6.2)
J	How do the deployment models agree?	Two-learner mean (§5.2)
K	What do the full comparisons and tests show?	Ranking claims and tests (§5.2, §6.1)
L	How does DiMTE use pseudo-labels?	Few and zero labels (§6.2)
M	How does ranking scale with the library?	Ranking across simulators (§6.1)
N	Does a learned embedding space help?	None, it tests an alternative design
O	Does DiMTE rank camera-trap locations?	Other domains (§6.3)
P	Does DiMTE rank source languages?	Other domains (§6.3)
Q	Does DiMTE rank sources for a third task?	UL-AoA columns of Table 1
R	How did the protocol change during the study?	DeepMIMO test cities corroborate (§5.1)
S	What does each term mean?	Every section

A FORMAL RESULTS AND DIAGNOSTICS

This appendix states the source summary in full and proves Proposition 1. It also gives the fixed-sample weighted-F1 bound of Section 3 and the error floor of symmetric scores, and it measures the mismatch between the proxy and a trained deployment model.

A.1 THE SOURCE SUMMARY IN FULL

For each active source class $c \in \mathcal{Y}_s$, that is, each class that occurs in the source, k-means produces $M_{sc} \leq \min(M, n_{sc})$ prototypes μ_{scj} and weights $w_{scj} = n_{scj}/n_{sc}$. Here n_s counts the source samples, n_{sc} those of class c, and n_{scj} those assigned to prototype j. Add-one class frequencies can be normalized over active source classes without changing the decision rule,

$$\pi_{sc} = \frac{n_{sc} + 1}{n_s + |\mathcal{Y}_s|}. \tag{6}$$

Let $v_{sc} \in \mathbb{R}^k$ be the coordinatewise mean squared residual around the assigned prototypes and let $\bar{v}_s$ be the coordinatewise variance of all source embeddings. We shrink each class variance toward $\bar{v}_s$ and pool across classes,

$$\sigma_s^2 = \frac{1}{n_s} \sum_{c \in \mathcal{Y}_s} n_{sc} \frac{n_{sc} v_{sc} + \kappa \bar{v}_s}{n_{sc} + \kappa}, \qquad \Sigma_s = \operatorname{diag}\big(\max\{\sigma_s^2, 10^{-4}\}\big), \tag{7}$$

with $\kappa = 10$ and the maximum applied coordinatewise. The proxy logits of (4) then read these quantities.

Terminology. As in the main text, this appendix calls $\widehat{P}_{s \to t}$ an estimate. It is a directed estimate of the target risk of a specified deployment model, not a mathematical distance.

A.2 Uniform Proxy-Risk and Selection-Regret Bounds

We first prove Proposition 1. For a fixed source s, write

$$R_t(g_s) = \mathbb{E}_{P_t}\bar{\ell}_s(x, y, a), \qquad R_t(\mathcal{A}) = T_{\mathcal{A}}(s \to t).$$

By the triangle inequality,

$$\begin{aligned}\left|\widehat{P}_{s\to t} - R_t(\mathcal{A})\right| &\le \left|\widehat{P}_{s\to t} - R_t(g_s)\right| + |R_t(g_s) - R_t(\mathcal{A})| \\ &\le \left|\widehat{P}_{s\to t} - R_t(g_s)\right| + \Delta_{s,t}.\end{aligned}$$

Because the representation, preprocessing, and source proxy are fixed independently of the labeled target sample, Hoeffding's inequality gives

$$\Pr\left[\left|\widehat{P}_{s\to t} - R_t(g_s)\right| > B\sqrt{\frac{\log(2N/\delta)}{2m}}\right] \le \frac{\delta}{N}.$$

A union bound over the N candidate sources proves (5).

Let $\varepsilon = B\sqrt{\log(2N/\delta)/(2m)}$. On the event above,

$$\begin{aligned}T_{\mathcal{A}}(\hat{s} \to t) &\le \widehat{P}_{\hat{s}\to t} + \Delta_{\hat{s},t} + \varepsilon \\ &\le \widehat{P}_{s^*\to t} + \Delta_{\hat{s},t} + \varepsilon \\ &\le T_{\mathcal{A}}(s^* \to t) + \Delta_{s^*,t} + \Delta_{\hat{s},t} + 2\varepsilon,\end{aligned}$$

which is the selection-regret bound of Proposition 1. For a hard proxy, if the loss depends on a classifier only through its predicted class and lies in $[0, B]$, then the two losses agree whenever the predictions agree, hence

$$\Delta_{s,t} \le B\,\mathbb{E}_{\xi}\Pr_t[g_s(x) \ne h_{s,\xi}(x)].$$

The same argument orders pairs of sources. For a fixed pair (a, b), apply Hoeffding's inequality to the two sources alone. With probability at least $1 - \delta$, $|\widehat{P}_{a\to t} - T_{\mathcal{A}}(a \to t)| \le \Delta_{a,t} + \varepsilon_2$, and the same holds for b, where $\varepsilon_2 = B\sqrt{\log(4/\delta)/(2m)}$. So whenever

$$|T_{\mathcal{A}}(a \to t) - T_{\mathcal{A}}(b \to t)| > \Delta_{a,t} + \Delta_{b,t} + 2\varepsilon_2, \tag{8}$$

the difference $\widehat{P}_{a\to t} - \widehat{P}_{b\to t}$ has the sign of $T_{\mathcal{A}}(a \to t) - T_{\mathcal{A}}(b \to t)$, and the estimate orders the pair correctly. The statement holds for all pairs of a target at once when $\log(2N/\delta)$ replaces $\log(4/\delta)$. The *resolved* column of Table 4 is the share of pairs that meet (8) for one training seed of the raw-CSI CNN. The measured risk of that CNN stands in for $T_{\mathcal{A}}$, and its disagreement with the proxy bounds Δ as above. The column shows how far the bound is from ordering sources. It certifies nothing, because the ray-traced targets are not i.i.d. and the mismatch term needs the trained model.

Proposition 1 is a superpopulation statement for i.i.d. labeled target examples. The ray-traced users are sampled from spatial grids, so neighboring users are dependent, and the proposition is not a finite-sample certificate for those grids. The conditional independence used above also fails when the preprocessing map is fitted on the same target examples that enter the empirical risk. Appendix F therefore also reports a map fitted on samples of the training cities alone.

Two signals flag a large mismatch before any deployment model exists, on LOS/NLOS identification. The mismatch $\Delta_{s,t}$ needs a trained deployment model, so we test two signals that need none. The first is the disagreement among three source classifiers of Table 29, namely the sum of the absolute differences between the mixture estimate and the k-NN and Parzen estimates. The second is the share of target samples whose Mahalanobis distance to the nearest source component exceeds the 95th percentile of the source's own samples. On LOS/NLOS identification in the Sionna test cities, the tenth of cross-city pairs with the most disagreement has an absolute error of 0.080 in $1-$F1, against 0.045 for the other pairs. On the DeepMIMO test cities the two errors are 0.091 and 0.041. The share outside the support separates the pairs as well, with 0.124 against 0.040 and 0.096 against 0.040. On beam prediction only the disagreement flags larger errors, and only on the Sionna test cities (6.4 against 4.2 dB), so neither signal is a reliable warning there.

A.3 Weighted-F1 Stability on a Fixed Sample

Weighted F1 is not an average of per-example losses, so Proposition 1 does not apply to it directly. The following deterministic statement explains how disagreement between the proxy and the deployment model affects it on a fixed target sample.

Proposition 2 (Fixed-sample weighted-F1 stability)**.** *Let classifiers g and h predict labels for the same n examples and disagree on $k = \epsilon n$ of them. Then*

$$|\mathrm{F1_w}(g) - \mathrm{F1_w}(h)| \le 4\epsilon.$$

Proof. Let n_c be the number of examples with true class c, m_c the number predicted as c, and T_c the correct predictions of class c, so $F_c = 2T_c/(n_c + m_c)$ and $\mathrm{F1_w} = \sum_c (n_c/n)F_c$. Let a_c of the changed examples have true class c, and let b_c count changed examples for which either classifier predicts c. Then $\sum_c a_c = k$ and $\sum_c b_c \le 2k$. Since $|\Delta T_c| \le a_c$, $|\Delta m_c| \le b_c$, $2T_c \le n_c + m_c$, and both denominators are at least n_c,

$$|F_c(g) - F_c(h)| \le \frac{2a_c + b_c}{n_c}.$$

Weighting by n_c/n and summing gives at most $(2k + 2k)/n = 4\epsilon$. □

The soft-confusion score used by DiMTE is a plug-in surrogate for weighted F1. The proposition above applies to hard predictions and does not make that plug-in score an unbiased estimate of expected F1. The plug-in does not drive the ranking. The accuracy readout, an additive loss that the proxy estimates without bias, ranks LOS/NLOS sources at 0.83 on the DeepMIMO test cities and 0.79 on the Sionna test cities. The plug-in reaches 0.85 and 0.78, and the two estimates order the cross-city pairs with rank correlations of 0.98 and 1.00.

A.4 An Error Floor for Symmetric Scores

Because transfer is directed, a symmetric score cannot reproduce an asymmetric transfer matrix.

Proposition 3. *Let Π be a set of ordered source-to-target pairs closed under reversal. Every symmetric predictor $f(s,t) = f(t,s)$ satisfies*

$$\frac{1}{|\Pi|} \sum_{(s,t)\in\Pi} |f(s,t) - P_{st}| \ge \frac{1}{2|\Pi|} \sum_{(s,t)\in\Pi} |P_{st} - P_{ts}|. \tag{9}$$

Proof. Group the ordered pairs into unordered pairs $\{s,t\}$. For one such pair, symmetry gives $f(s,t) = f(t,s) = f$, and the triangle inequality gives $|f - P_{st}| + |f - P_{ts}| \ge |P_{st} - P_{ts}|$. Summing over unordered pairs and dividing by $|\Pi|$ proves (9). Equality holds whenever $f(s,t)$ lies between P_{st} and P_{ts} for every unordered pair, and the midpoint is one such choice. □

The same geometry constrains bidirectional rankings. For a target t, let a, b, and c be the centered, normalized rank vectors of the transfer column $\mathbf{P}_{\cdot t}$, transfer row $\mathbf{P}_{t\cdot}$, and symmetric score vector $f(\cdot,t) = f(t,\cdot)$. Spearman correlation is the inner product of these rank vectors, and angular distance on the unit sphere satisfies the triangle inequality. Therefore

$$\arccos\rho(c,a) + \arccos\rho(c,b) \ge \arccos\rho(a,b).$$

The inequality states a limitation of symmetric scores, not a guarantee that any directed estimator ranks well.

A.5 Measured Mismatch Between Proxy and Deployment Model

We measure the otherwise unobservable mismatch term by training the raw-CSI CNN on each source and comparing its predictions with those of the source proxy on every target, for LOS/NLOS identification in the training cities. With one component per class, the classifiers disagree on 0.094 of cross-city target examples and the proxy-risk error is 0.072 on average. With 32 components, disagreement falls to 0.092 and error to 0.064. Pairwise disagreement then correlates with estimation error at $\rho = 0.89$. Table 4 repeats the diagnostic for both tasks on both test sets. The diagnostic supports the mechanism of Proposition 1, but it is retrospective, because it needs the trained CNN.

Table 4: Proposition 1 measured on the test sets. For every cross-city pair, g_s is the hard classifier of the source's summary and h_s the raw-CSI CNN trained on the source (seed 0), both evaluated on the target samples that DiMTE reads. The loss is the classification error on LOS/NLOS identification and the L1-RSRP difference in dB on beam prediction. *Rate* is the share of samples on which g_s and h_s disagree and *loss gap* the mean $|\ell(g_s) - \ell(h_s)|$ on those samples. The bound has two terms, averaged over pairs: (i) the *mismatch* term L times the rate, with L the target's largest per-sample loss (1 on LOS/NLOS identification, and a median of 89 dB and 97 dB on beam prediction), and (ii) the *sampling* term $L\sqrt{\log(2N/\delta)/(2m)}$, with N candidate sources, $m = 1000$ target samples and $\delta = 0.05$. *Error* is the absolute difference between the mean losses of g_s and h_s on the same samples, averaged over pairs, which is what the bound covers. *Resolved* is the share of a target's source pairs whose order the pairwise condition of Appendix A.2 guarantees. It is a diagnostic rather than a certificate, because the ray-traced users are not i.i.d. and the mismatch term needs the trained model. *Tracking* is the Spearman correlation over pairs between the rate and the error.

	Disagreement		Bound of Proposition 1					
Test set	rate	loss gap	mismatch	sampling	total	Error	Resolved	Tracking
LOS/NLOS, classification error								
DeepMIMO	10.4%	1.000	0.104	0.064	0.168	0.050	12.6%	+0.73
Sionna	7.9%	1.000	0.079	0.064	0.142	0.044	8.0%	+0.76
Beam prediction, L1-RSRP difference in dB								
DeepMIMO	54.1%	12.3	48.4	5.7	54.1	3.3	0.4%	+0.25
Sionna	57.0%	14.0	56.0	6.3	62.3	4.4	0.0%	+0.49

B Transfer-Matrix Noise and Asymmetry

A transfer matrix is a measurement, and its noise sets the floor for every claim built on it. This appendix measures that noise and the asymmetry of the measured transfer, which Sections 4.3 and 5.2 use. Each matrix averages three training seeds, each with its own within-dataset split. Fine-tuned LWM on beam prediction has two seeds in the training and DeepMIMO test cities. Table 5 reports six quantities for each benchmark: (i) the mean transfer metric over cross-city pairs, (ii) the replicate discrepancy, that is the mean absolute difference between two seeds' matrices, (iii) the mean asymmetry $|P_{st} - P_{ts}|$ of the averaged matrix, (iv) the ratio of (iii) to (ii), (v) the share of cross-city pairs whose asymmetry exceeds two standard errors of its estimate, and (vi) the mean over datasets t of the Spearman correlation between the column $\mathbf{P}_{\cdot t}$ (how well other datasets transfer to t) and the row $\mathbf{P}_{t\cdot}$ (how well t transfers to them).

Because the matrix is estimated, no score can be expected to match it perfectly. The reliability column of Table 18 is a practical level for comparison rather than a hard bound, because a score that predicts the mean of the seeds can exceed the agreement of a single seed with that mean. One training seed of either deployment model of the two-learner mean predicts the per-target order of the other seeds at $\rho = 0.82$ to 0.84 on LOS/NLOS identification and 0.81 to 0.82 on beam prediction. The two deployment models predict each other at 0.84 and 0.74. The per-target correlations of Table 21 should be read against these numbers. For the matrices of the MLP on frozen tokens, the same level was 0.70, 0.83 and 0.91 on the three benchmarks of the earlier protocol.

The transfer matrix also depends on the deployment model. Section 5.2 measures it for four deployment models. On LOS/NLOS identification, the matrix of the MLP on frozen LWM tokens agrees with the two deployment models of the two-learner mean at a per-target ρ of 0.70 on average. Those two agree with each other at 0.84. Any score therefore predicts transfer for a stated deployment model, and this paper states it.

Table 5 supports two conclusions. The asymmetry is real, since for the two-learner mean it is 2.2 to 5.0 times the replicate noise and exceeds two standard errors for 72% to 86% of pairs. Each standard error pools two or three seeds of each of the two deployment models. Under a Student t distribution with two to four degrees of freedom, two standard errors correspond to a one-sided confidence of about 91% to 94%. The share of pairs above two standard errors therefore describes the matrix as a whole rather than certifying single pairs. The asymmetry also matters for selection. A dataset's best sources and its best targets are unrelated orderings on LOS/NLOS identification (row–column ρ of 0.03), and only partly related on beam prediction (0.47). Seeds agree cell by cell at 0.90 to 0.94 (the per-target reliability of Table 18 is a stricter measure), so three seeds are enough for the averaged matrix to serve as the measured transfer. A single seed was not (Appendix R). On the DeepMIMO test set the asymmetry has about the same size (0.165 against 0.154 in $1-$F1, 3.6 against 3.7 dB).

Table 5: Noise and asymmetry of the seed-averaged transfer matrices. The rows cover the two-learner mean on the training cities, the DeepMIMO test cities and the Sionna test cities, and the MLP on frozen LWM tokens, the deployment model of the earlier protocol, on the training cities. Each dataset has $n = 1000$ training samples unless stated. The three task-metric columns give the mean transfer over cross-city pairs, the replicate discrepancy between the matrices of two seeds, and the mean asymmetry $|P_{st} - P_{ts}|$. Ratio divides the mean asymmetry by the replicate discrepancy, computed from unrounded values, and the next column gives the share of pairs whose asymmetry exceeds two standard errors. Seed is the per-target ρ between one training seed and the mean of the others, as in Table 18. A symmetric score treats the sources and the targets of a dataset identically, so the row–column correlation r of the last column limits it. A symmetric score that ranks the sources at ρ_c ranks the targets at most at $\cos(\arccos \rho_c - \arccos r)$, and one that ranks both equally reaches at most $\sqrt{(1+r)/2}$ on both (Appendix A). The two-learner rows average the fine-tuned LWM and the raw-CSI CNN, and their replicate discrepancy comes from each deployment model's own seeds. Their beam rows use the 8-antenna input, while the earlier protocol used the full array.

	Task metric			Asymmetry		Spearman ρ	
Deployment model and set	Mean	Replicate	$\lvert P_{st} - P_{ts}\rvert$	Ratio	Pairs >2 SE	Seed	Row–column
LOS/NLOS identification, 1−F1							
MLP on frozen tokens, n=400	0.077	0.024	0.083	3.5	68%	–	0.27
MLP on frozen tokens, n=1000	0.089	0.022	0.101	4.6	79%	–	0.22
Two-learner mean, training	0.124	0.031	0.154	5.0	86%	0.83	0.03
Two-learner mean, DeepMIMO	0.180	0.034	0.165	4.9	83%	0.91	0.22
Two-learner mean, Sionna	0.102	0.032	0.113	3.6	78%	0.80	0.17
Beam prediction, L1-RSRP difference in dB							
MLP on frozen tokens, full array	18.9	1.3	3.3	2.6	70%	–	0.57
Two-learner mean, training	13.3	1.7	3.7	2.2	72%	0.81	0.47
Two-learner mean, DeepMIMO	10.6	1.2	3.6	2.9	79%	0.90	0.60
Two-learner mean, Sionna	15.7	1.7	4.5	2.6	76%	0.89	0.48

Rows and columns agree more there (row–column ρ of 0.22 and 0.60), so that set is not more directed than the training one.

Neither the seed noise of the measured transfer nor the target samples that DiMTE reads explain the margins. The paired tests of the main text treat the seed-averaged matrix as fixed. We therefore redraw the training seeds of each deployment model 500 times with replacement and score every method against each redrawn matrix. The correlation of DiMTE then has a standard deviation of at most 0.011, and its lead over every comparator of the full tables stays positive in every redraw. The sign changes only in two near-ties, for LOS/NLOS identification on the Sionna test cities, in at most 37.6% of the redraws. Both are estimators that read a trained model per source, LEEP on the raw-CSI CNN and NCE on fine-tuned LWM. DiMTE reads each target's seed-0 training split, and the later seeds redraw that split. Only the seed-0 test samples are therefore disjoint from the samples that DiMTE reads (Appendix J.1). Against the matrix of seed 0 of both deployment models alone, the margin over the strongest comparator is +0.03 and +0.11 on the Sionna test cities and +0.12 and +0.13 on the DeepMIMO test cities. These margins lie within 0.01 of the margins against the full matrix, and every one-sided p is below 0.001.

C The Wireless Datasets

This appendix shows what a wireless dataset contains, for readers who do not work with radio channels. A dataset, or site, is the set of users that one base station serves in one city. The ray tracer places users on a 1 m grid, 1.5 to 2 m above the ground, and traces the propagation paths from the base station to each of them. Each path has a gain, a delay and a direction. DeepMIMO sums the paths into the channel between the base station's 32-antenna array and the user's single antenna, and this channel is one sample. Its LOS/NLOS label states whether the direct path exists, and its beam label is the strongest of the 32 beams of the codebook, a fixed set of discrete Fourier transform (DFT) beams (Section 5). Users without any path receive no signal and are left out. Table 6 summarizes the three sets, and Figure 3 shows one dataset of each, the one with the set's median LOS share.

The propagation state and the beam label follow the street layout of a site. Users on streets that the base station sees are LOS, and users in side streets receive only reflected or diffracted paths. The beam label turns smoothly with the direction of LOS users. In NLOS areas it jumps wherever the

Table 6: The three wireless sets. Users with a path are counted over all users of a dataset and given as the median (range) over its datasets. The LOS share and the number of beams that label at least 1% of the users are medians over datasets, computed on each dataset's 1000 training samples.

Set	Ray tracer	Cities	Datasets	Users with a path, median (range)	LOS share	Beams in use
Training cities	Wireless InSite	44	68	26,281 (7,068–108,403)	0.36	21
DeepMIMO test cities	Wireless InSite	59	84	60,675 (5,972–2,538,080)	0.76	22
Sionna test cities	Sionna RT	42	84	21,715 (3,811–61,219)	0.39	18

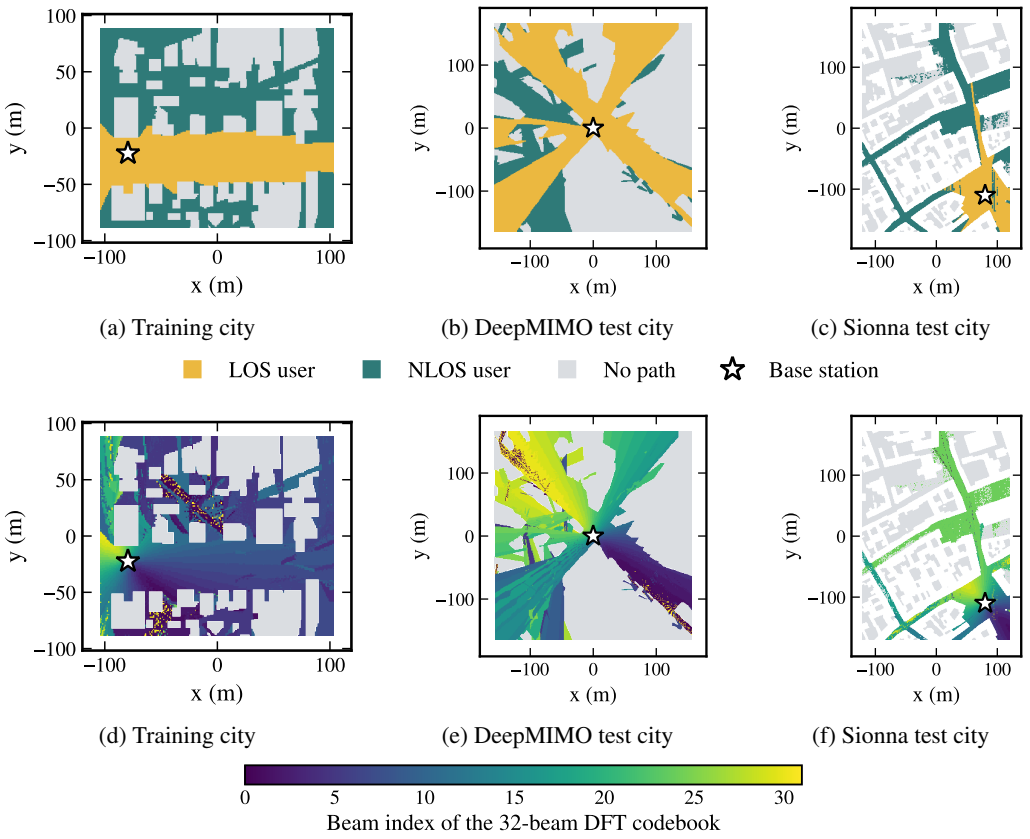
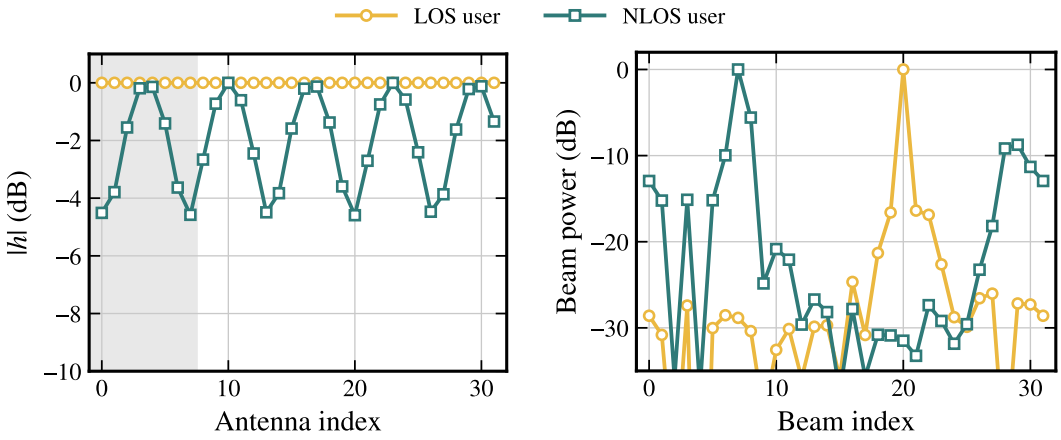


Figure 3: One dataset of each set on its 1 m user grid, with the base station as a star, in San Diego (a,d), La Habana (b,e) and Copenhagen (c,f). (a)–(c) The propagation state of every user. (d)–(f) The beam label of every user with a path. Gray users receive no path. In the DeepMIMO cities they include the users inside buildings, while the Sionna test cities place no users there (white). The three datasets have 22,633, 55,158 and 18,629 users with a path, of which each seed draws 1000 for training.

strongest reflection changes, which produces the speckled regions of Figures 3d and 3e. So two sites differ in which beams occur, in how often each occurs, and in what the channels of each beam look like. The class priors and the per-class prototypes of DiMTE describe these three differences.

A LOS channel is close to one plane wave, while an NLOS channel mixes several paths. Figure 4 shows a LOS and an NLOS user of the La Habana site of Figure 3b, each at the median distance of its state from the base station (127 and 129 m). The LOS user receives one dominant path, so its amplitude is the same on every antenna and its beam power has one sharp peak. The NLOS user receives several paths of comparable strength. The paths interfere across the array, so its amplitude ripples by several dB and its power spreads over two groups of beams. The beam-prediction model observes only the first eight antennas, shaded in Figure 4a, and must infer the best beam of the full array from them.



(a) Amplitude across the array

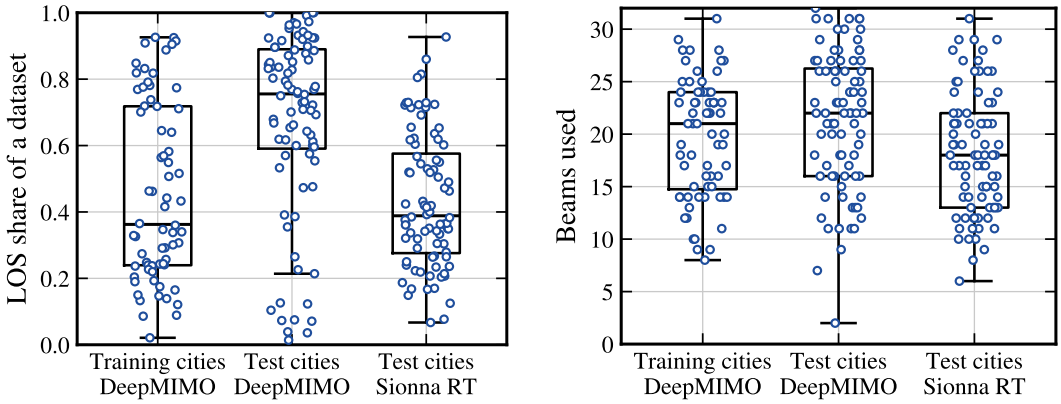
(b) Power of the 32 DFT beams

Figure 4: The channels of a LOS and an NLOS user of the La Habana site of the DeepMIMO test cities. (a) The amplitude $|h|$ on each antenna, relative to the user's largest value. The beam-prediction model observes the shaded eight antennas. (b) The power that each DFT beam collects, relative to the strongest beam, whose index is the beam label.

Each sample holds one narrowband channel on 32 subcarriers, the input format of LWM. A narrowband channel suits the two tasks, which depend on the spatial signature of the site.

The encoder has seen channels of some training cities and of no test city. The LWM checkpoint we use was pretrained by masked channel modeling, which reconstructs hidden parts of a channel and reads no label. Its pretraining data are DeepMIMO channels of 20 cities, the ASU campus, Boston 5G and the O1 scenario. All 20 cities belong to our 44 training cities. None belongs to the DeepMIMO or Sionna test cities, so no test channel shaped the representation.

The three sets differ in their propagation mix (Table 6 and Figure 5). The DeepMIMO test set is mostly LOS, with a median LOS share of 0.76 per dataset, while the training and Sionna test sets are mostly NLOS, at 0.36 and 0.39. The median dataset uses 21, 22 and 18 of the 32 beams in the three sets. This difference in propagation is one reason why results are compared within a set.



(a) LOS share

(b) Beams in use

Figure 5: Per-dataset statistics of the three sets, on each dataset's 1000 training samples. (a) The share of users with a LOS path. (b) The number of the 32 beams that label at least 1% of the users. Each marker is one dataset. Boxes span the quartiles, with the median as a line, and whiskers reach the most extreme dataset within 1.5 times the interquartile range.

D THE SIONNA TEST CITIES

Every DeepMIMO city at 3.5 GHz belongs to the training or the DeepMIMO test set, so a test on sites that no design choice has seen needs new cities. We chose 45 dense city centers outside DeepMIMO on five continents before generating any channel. The generation follows the DeepMIMO city scenarios in four steps: (i) building footprints come from OpenStreetMap, (ii) one or two base stations are placed on open streets, (iii) Sionna RT (Aït Aoudia et al., 2025) traces the paths to a grid of users, and (iv) the DeepMIMO code builds the channels and the labels. The footprints lie within 300 m by 400 m of the center, courtyards included. They are extruded to their tagged height, with 3 m per level and 10 m when untagged, as in the OpenStreetMap scenarios of DeepMIMO. The base stations stand 10 m above the ground on open street points at least 3 m from any building. The first is nearest the center, and the second is at least 60 m away. Each must reach at least 20% of 1000 sparse outdoor points with a path, so that no station stands in an enclosed courtyard. Users lie on a 1 m grid 1.5 m above the ground over 240 m by 340 m. Sionna RT traces up to four interactions, with specular reflections and diffraction, using the ITU material models of concrete for the buildings and of wet ground, at 3.5 GHz and with isotropic antennas. The paths are converted to the DeepMIMO format, and the unchanged DeepMIMO code of the other sets builds the channels (32 antennas rotated by -135° and 32 subcarriers, narrowband as in Appendix C) and the labels. A base station is kept when at least 2600 users receive a path, as for the DeepMIMO test set. Sionna RT differs from Wireless InSite, the ray tracer behind DeepMIMO, in its ray launching and diffraction model, so the set also tests a change of simulator. Three cities had too few mapped buildings and were skipped. That leaves 84 datasets in 42 cities, with a median of 21,715 users per base station and LOS shares from 0.06 to 0.91 (Table 7). Transfer is measured exactly as for the other sets, and Table 1 scores every method on the result.

In the Top-1/1 metric of TR 38.843, the selected sources of the best rules are within a few points of each other. On this set the source that DiMTE selects reaches 43% Top-1/1 accuracy, the source of DiMTE (learned space) 45% and that of the best baseline 47%, against 29% for a random source and 50% for the genie-aided source. The L1-RSRP difference separates the rules more than the accuracy does, because a wrong beam next to the best one costs little power but counts as a full error in accuracy.

Table 7: The Sionna test set, cities outside DeepMIMO ray traced with Sionna RT (Appendix D). For each kept base station, the number of users that receive a path and the LOS share. Three cities were tried but skipped for too few OpenStreetMap buildings, namely Budapest, Busan and Hanoi.

City	Users with a path	LOS share	City	Users with a path	LOS share
Atlanta	38,424, 25,079	0.38, 0.54	Mexico City	10,883, 15,745	0.36, 0.39
Auckland	17,492, 7,880	0.24, 0.22	Milan	21,305, 33,905	0.19, 0.57
Baltimore	20,501, 16,212	0.32, 0.16	Minneapolis	27,219, 15,608	0.22, 0.07
Berlin	26,808, 20,868	0.30, 0.71	Nagoya	35,666, 38,860	0.31, 0.67
Boston	10,445, 5,426	0.22, 0.42	Naples	15,590, 10,243	0.64, 0.17
Brisbane	8,441, 9,876	0.71, 0.44	Oslo	22,918, 19,385	0.41, 0.42
Bucharest	21,966, 34,217	0.33, 0.55	Paris	36,002, 23,275	0.49, 0.43
Buenos Aires	39,253, 26,717	0.60, 0.36	Perth	4,911, 7,874	0.85, 0.62
Copenhagen	14,381, 18,629	0.18, 0.40	Pittsburgh	12,454, 10,414	0.33, 0.53
Dublin	26,850, 10,901	0.73, 0.13	Portland	32,599, 24,513	0.37, 0.30
Frankfurt	15,127, 18,596	0.30, 0.71	Porto	26,121, 30,074	0.73, 0.69
Glasgow	21,719, 9,217	0.46, 0.23	Rotterdam	18,058, 15,046	0.24, 0.35
Hamburg	25,881, 25,962	0.26, 0.36	Sapporo	28,603, 40,916	0.27, 0.67
Ho Chi Minh City	40,371, 49,689	0.37, 0.52	Seville	6,332, 20,866	0.27, 0.72
Krakow	32,830, 28,536	0.59, 0.61	Sofia	36,423, 31,967	0.32, 0.35
Kuala Lumpur	31,895, 28,986	0.51, 0.16	Tel Aviv	8,700, 20,020	0.06, 0.25
Lima	22,910, 22,770	0.34, 0.59	Tunis	59,334, 61,219	0.72, 0.52
London	31,977, 27,150	0.79, 0.91	Valencia	30,190, 24,371	0.71, 0.82
Lyon	15,552, 16,453	0.34, 0.19	Vancouver	32,973, 20,790	0.55, 0.44
Manchester	4,172, 10,001	0.38, 0.18	Vienna	23,685, 17,617	0.45, 0.37
Marseille	21,712, 18,640	0.54, 0.24	Zurich	3,811, 19,808	0.26, 0.53

E IMPLEMENTATION, BASELINES, AND COMPUTE

This appendix gives the recipes of the four deployment models, the interval method, the settings of every baseline and of DiMTE, and the compute of each method.

All four deployment models share one protocol of splits and seeds. Every model trains on 1000 samples of a source, selects its epoch on 500 validation samples of that source, and is tested on 1000 samples of each target. The splits are redrawn with each seed. The two deployment models of the two-learner mean have three seeds on every set, except fine-tuned LWM on beam prediction, which has two on the training and DeepMIMO test cities. The MLP on frozen tokens was measured on the training and DeepMIMO test cities, and the linear classifier on LOS/NLOS identification in the training cities (Table 18). Their recipes were chosen by in-site validation on four to six training datasets, before any transfer outcome was read. The linear classifier standardizes the frozen `[CLS]` embedding with source statistics (Adam, learning rate 10^{-2}, 100 epochs). The MLP on frozen tokens reads the concatenated patch-token embeddings. On LOS/NLOS identification it has layers of width 64 and 32 with batch normalization, ReLU and dropout, and on beam prediction layers of width 256, 128 and 64 with ReLU and dropout. It trains with Adam at learning rate 10^{-3} and batch 128 and selects its epoch by validation weighted F1. Fine-tuned LWM trains the encoder and the same head end to end with AdamW, weight decay 10^{-4}, batch 64, a cosine schedule and mixed precision. The encoder learning rate is 10^{-4} for 20 epochs on LOS/NLOS identification and 10^{-3} for 100 epochs on beam prediction, and the head learning rate is 10^{-3}. The raw-CSI CNN takes the real part, the imaginary part and the magnitude of the CSI as three channels. The three channels are divided by one fixed scale per source model, the median root-mean-square of the source's training CSI, so that the received power is preserved. Per-sample normalization discarded the received power and made the CNN weak. The CNN has two blocks of two 3×3 convolutions (32 and 64 channels) with pooling. For LOS/NLOS identification these are followed by a 128-channel convolution with global average pooling (AdamW, learning rate 10^{-3}, 60 epochs). For the 8-antenna beam task they are followed by a flattened 128-unit layer, with pooling along subcarriers only (learning rate 3×10^{-3}, 120 epochs).

The prediction intervals of Appendix K.1 use conformalized quantile regression (Romano et al., 2019) and a weighted conformal correction (Tibshirani et al., 2019). A target sample is outside the support of a source summary when its Mahalanobis distance to the nearest source component exceeds the 95th percentile of the source's own samples. The support score, the share of target samples outside the support, reads 200 target samples per pair. The quantile regressors are gradient-boosted trees (200 trees of depth 2, learning rate 0.05, subsample 0.8) at the 5th and 95th percentiles. The conformal correction is the $(1-\alpha)$ empirical quantile of the calibration conformity scores, at $\alpha = 0.1$. Under shift, calibration pairs are weighted by the likelihood ratio of a boosted classifier that separates calibration from test inputs, clipped to $[10^{-3}, 1 - 10^{-3}]$, and the correction is the weighted quantile. That weighting assumes that the inputs shift while the map from score to metric stays fixed. A new city can break the second assumption, because its geometry changes how a given score translates into transfer. No reweighting of the inputs can repair such a change, which is one reason the intervals under-cover.

The baselines fall in five groups: (i) label-only rules, (ii) dataset scores, (iii) transferability estimators, (iv) a self-supervised set encoder, and (v) set encoders trained on the transfer matrix. The label-only rules are the total-variation and the directed KL distance between class priors. The dataset scores are label-aware centroid and sliced-Wasserstein (SW) distances on raw channels and on LWM embeddings, and the directed joint KL divergence on both. That divergence adds the KL divergence of the class priors to the prior-weighted KL divergence of per-class diagonal Gaussians in 32 principal coordinates. The dataset scores also include DiMTE with one component on raw channels, label-free SW and maximum mean discrepancy (MMD) (Gretton et al., 2012) distances, label-aware MMD, centered kernel alignment (CKA) (Kornblith et al., 2019), OTDD (Alvarez-Melis & Fusi, 2020), Task2Vec (Achille et al., 2019) and UMAP projections (McInnes et al., 2020). The transferability estimators follow the protocol of that literature and score a model trained per source, the MLP on frozen LWM tokens of Section 5.2. They are LEEP, NCE, N-LEEP, LogME, regularized H-score, GBC, TransRate and PARC. LEEP, NCE and OTCE also run on the same lightweight source proxy as DiMTE, with and without standardization, and LEEP and NCE on its mixture summary with the task's k and M. LEEP and NCE also run on both proxies in the projection of DiMTE, together with two directed analytical scores, JC-NCE (Tan et al., 2024) with the OTDD ground cost and WDJE (Zhan et al., 2026) with our mixture as its source model. The self-supervised Deep Sets

encoder trains without any transfer matrix. The last group receives the same library transfer matrix as DiMTE (learned space). It holds a pairwise regressor from dataset descriptors to P_{st} in the LWM and raw spaces, directed Deep Sets (Zaheer et al., 2017) encoders on LWM embeddings or raw channels, and a directed Set Transformer (Lee et al., 2019), all trained with the loss of Appendix N. Supervised methods use the same inner model selection and five-seed averaging as DiMTE (learned space). Beam prediction adds five scores that read the target's L1-RSRP sweep (Table 8).

Table 8 gives the representation and settings of every baseline and says which of them, if any, were tuned.

Table 8: Settings of every baseline. A baseline runs at the settings of its paper or at the fixed values listed, with no search on a test set, except where the last column names what was tuned: (i) LEEP and NCE on our mixture borrow our k and M, in either projection, (ii) the learned models choose their number of epochs by an inner three-fold split of the training cities and average five seeds, as DiMTE (learned space) does, and (iii) the scores that read the beam sweep are reported at the best of three temperatures on each test set, which favors them. Linear CKA between two sets of two centered class centroids is always one, so CKA ranks LOS/NLOS sources only where a dataset lacks a class, as three DeepMIMO test datasets do. CKA is therefore among the 49 LOS/NLOS comparators of the DeepMIMO test cities and not among the 48 of the Sionna test cities. Beam prediction adds the five scores that read the sweep, for 54 comparators on the Sionna test cities. The closed-form proxies fit their principal directions on the training cities, except in the rows marked *our projection*, which fit them on the samples of the scored set as DiMTE does.

Method	Representation and settings	Tuned
Label-only rules		
Class-prior distances	total variation or directed KL of class frequencies	–
Dataset scores, no model per source		
Label-aware centroid and SW	raw or LWM, 64 SW projections, 32 quantiles	–
Label-free SW	raw channels, classes ignored	–
Joint KL, directed	32 principal coordinates, diagonal Gaussians	–
DiMTE with one component	raw channels	–
MMD, label-free and label-aware	LWM, RBF kernel, median-heuristic bandwidth	–
CKA on class centroids	LWM, one minus linear CKA	–
OTDD	raw channels, 200 samples per dataset	–
UMAP, 2-d, 16-d and supervised 2-d	raw, 400 samples, 16 neighbors, $d_{\min}$=0.1	–
Task2Vec	32-d LWM or 64-d raw PCA, logistic probe	–
Estimators on a closed-form source proxy		
LEEP, NCE and OTCE	32 principal coordinates, tied Gaussian, κ=10	–
The same, standardized	standardized principal coordinates	–
LEEP and NCE on our mixture	our mixture summaries	our k, M
LEEP and NCE, our projection	tied Gaussian, k=32	–
The same on our mixture	our mixture summaries	our k, M
JC-NCE	joint transport, OTDD cost, λ=0.5, 200 samples	–
WDJE	our mixture as source model, $k\lambda$=0.001	–
Estimators on a model trained per source		
LEEP, NCE, N-LEEP and LogME	MLP on frozen LWM tokens, seed 0	–
H-score, GBC, TransRate and PARC	the same models, published formulas	–
The eight, SFDA, NCTI, ETran, DISCO	fine-tuned LWM and raw-CSI CNN, seed 0	–
Learned set encoders		
Deep Sets, self-supervised	32 principal coordinates, InfoNCE, 300 epochs	–
Pairwise regressor	mean, spread and prior of LWM or raw features	epochs
Directed Deep Sets, Set Transformer	32 principal coordinates, ranking losses	epochs
Beam scores that read the L1-RSRP sweep		
Class-prior KL, joint KL, centroid	soft labels from the sweep, $T \in \{1, 3, 10\}$ dB	T, test set

Directed analytical scores rank sources far below DiMTE. JC-NCE matches source and target samples by optimal transport under the OTDD cost, which adds a class-conditional Wasserstein term with weight $\lambda = 0.5$ to the squared distance. It then scores the conditional entropy of the target labels given the matched source labels. WDJE bounds the target risk by the source model's own risk plus Wasserstein distances between the feature distributions and between the label distributions. We give it our mixture proxy as the source model and the proxy's cross-entropy on its own samples as that risk. Both scores run on 200 samples per dataset in the projection of DiMTE. On the Sionna test cities JC-NCE reaches 0.19 and 0.43 and WDJE 0.17 and 0.07, against 0.78 and 0.79 for DiMTE. On the DeepMIMO test cities JC-NCE reaches 0.33 and 0.54 and WDJE 0.25 and 0.67. The weight

$k\lambda = 0.001$ of the WDJE paper makes its feature term negligible. The best of 0.001, 0.1 and 1 on each test set, a choice that favors WDJE, reaches at most 0.70. Both scores measure how far two datasets lie apart or how their labels correspond, while the deployment metric rewards how a source's decision rule fares on the target.

Where prior methods sit. Table 9 places each family of prior methods on the four axes that separate it from DiMTE.

Table 9: Where prior methods sit on four axes: (i) *model*, whether they need a trained model for every candidate, (ii) *matrix*, whether they need a measured transfer matrix, (iii) *directed*, whether their score is directed, and (iv) *units*, whether it is a value in the units of the deployment loss. *Partly* marks frozen-feature probes, which return the accuracy of a probe fitted on target data, Task2Vec, which trains a probe per dataset, and the transfer-supervised predictors, which predict the metric where the other members of their family rank. DsDm needs many trainings, Taskonomy measures the matrix itself, and DiMTE reads the matrix of the training cities once per task, to choose k, M and τ, never when a new target is scored.

Family	Examples	Model	Matrix	Directed	Units
Model-based estimators	LEEP, NCE, LogME, SFDA	yes	no	yes	no
Label-conditional transport	OTCE, JC-NCE	yes	no	yes	no
Frozen-feature probes	Puigcerver et al., Renggli et al.	yes	no	yes	partly
Measured task transfer	Taskonomy	yes	is one	yes	yes
Model-aware data selection	DsDm	many	no	yes	yes
Cross-learning	Sun et al.	yes	no	no	no
Dataset distances	OTDD, MMD, SW, centroids	no	no	no	no
Task and dataset embeddings	Task2Vec, Dataset2Vec, Domain2Vec	partly	no	no	no
Directed divergences	Westny et al., directed KL	no	no	yes	no
Transfer-supervised predictors	Task2Box, LangRank, set encoders	no	yes	yes	partly
DiMTE (ours)	closed-form proxy per source	no	once	yes	yes

Table 10 lists every setting. DiMTE has three hyperparameters, namely the subspace size k, the softmax temperature τ and the number of components M. All three were chosen by the rank correlation on city-grouped folds of the training cities (Appendix J.1). The grids are $k \in \{16, 32, 64\}$ and $\tau \in \{0, 0.1, 0.3\}$, extended to $k = 128$ and $\tau = 1$ on beam prediction (Appendix F). The number of components then comes from $\{1, 4, 8, 16, 32, 64, 128\}$ with those settings (Appendix G). DiMTE (learned space) adds the optimizer settings and the loss temperatures, and none was tuned on test cities. Within-class variances are shrunk toward the dataset's pooled variance with weight $\kappa = 10$ before being tied across classes.

Table 10: Settings of DiMTE and DiMTE (learned space). The number of training epochs of DiMTE (learned space) is selected on validation cities by an inner three-fold split of the training cities and capped at 300.

Setting	Value
Embedding	ℓ_2-normalized LWM `[CLS]` token, $d = 128$
Subspace size k	32 (LOS/NLOS), 64 (beam), from 200 samples per dataset
Coordinates	standardized (DiMTE), raw (start of the learned variant)
Class-conditional model	M centroids per class (k-means), shared variance, smoothed prior
Components per class M	DiMTE 32 (LOS/NLOS) and 128 (beam), learned variant 1
Variance shrinkage κ	10
Softmax temperature τ	DiMTE 0.1 (LOS/NLOS) and 1.0 (beam), learned variant from 0.1
Prior weight β	1 (DiMTE), learned variant from 1
Residual network h	two layers, hidden width 256, GELU, zero-initialized output
Loss temperatures τ_d, τ_P	0.5, 0.3 (distances and losses standardized per batch)
Pairwise margin ξ	0.05 standard deviations of the training transfer metrics
Optimizer	AdamW, learning rate 3×10^{-4}, weight decay 10^{-3}, cosine decay
Weight averaging	exponential moving average, decay 0.98
Samples per step	all training samples of every training dataset
Epochs	selected on validation cities, at most 300
Seeds	5, distances averaged in metric units

Per new target and per new library, DiMTE costs CPU seconds, and its hyperparameters cost one training-city matrix per task. All experiments ran on one NVIDIA Quadro RTX 6000 (24 GB). LWM embeds 1000 samples in 0.11 and 0.03 s on the GPU and in 2.9 and 0.5 s on a CPU,

for LOS/NLOS identification and beam prediction. DiMTE scores all pairs of the 84 Sionna datasets in 15 and 161 s on a CPU. A model-based estimator first needs one trained deployment model per source. The training-city matrix that sets the hyperparameters of DiMTE trains every seed of both deployment models of the two-learner mean on the 68 training datasets, once per task (Table 11). This one-off cost exceeds that of one CNN per source for the library of all 236 datasets and is comparable to one fine-tuned LWM per source. For a library of N datasets and one new site, the measured transfer costs N trainings and N evaluations, or N evaluations if a trained model per source and task is already stored. DiMTE costs one embedding pass over the site's samples and N lightweight proxy evaluations.

E.1 Scaling Cost with the Library

For the library of all 236 datasets, DiMTE reads one new target against every source in 0.25 and 3.44 s on a CPU, and a new source adds 0.02 and 0.16 s. A model-based estimator first needs one trained model per source. For all 236 sources together, one seed of the raw-CSI CNN takes 0.29 GPU-hours on LOS/NLOS identification and 0.43 on beam prediction, and one seed of fine-tuned LWM 1.3 and 5.6. These totals scale the median training time of a few timed source models to the library, and the training repeats for every task and deployment model (Table 11). The hyperparameters k, M and τ of DiMTE come from the transfer matrix of the training cities. That matrix is measured once per task on its 68 datasets with every seed of both deployment models of the two-learner mean, which takes about 1.4 and 3.6 GPU-hours of training. The set encoders train on the same matrix. A fixed 1-NN rule needs no transfer matrix and stays within 0.03 of the tuned estimate on every set (Table 13).

Table 11: What ranking a library costs, for the library of all sets (236 datasets of 1000 samples), measured on an otherwise idle machine (one NVIDIA Quadro RTX 6000 GPU, with CPU time on 8 threads of an Intel Xeon Silver 4215R CPU @ 3.20GHz). The rows give the per-dataset embedding, the per-source summary and the per-target score of DiMTE, against the per-source training that the model-based estimators need before any scoring, with the recipes of the two deployment models. The last row is the transfer matrix of the training cities, measured once per task, on which the hyperparameters k, M and τ of DiMTE were chosen and the set encoders train. The summary grows with the number of components. At $M = 128$ on beam it keeps most samples, while at $M = 32$ on LOS/NLOS identification it stores 64 vectors per source. Asymptotically, embedding costs $O(Nn)$, the summaries $O(NnMk)$ per k-means iteration, and one target $O(Nn_tCMk)$, with N sources of n samples, n_t target samples, C classes, M components and k coordinates.

Step	LOS/NLOS	Beam prediction
LWM embedding, per 1000 samples, GPU	0.11 s	0.03 s
LWM embedding, per 1000 samples, CPU	2.9 s	0.5 s
Summary of one source, CPU	0.02 s	0.16 s
One target against every library source, CPU	0.25 s	3.44 s
Memory of the library's summaries	2.0 MB	56.4 MB
Memory of the library's samples, same coordinates	30.2 MB	60.4 MB
CNN per source, whole library (estimators)	0.29 GPU-h	0.43 GPU-h
Fine-tuned LWM per source, whole library (estimators)	1.3 GPU-h	5.6 GPU-h
Training-city transfer matrix, once per task	1.4 GPU-h	3.6 GPU-h

F Choice of the Subspace Size and the Temperature

This appendix shows how the subspace size k and the temperature τ were chosen on the training cities, and what a new task loses without that choice. Table 12 reports the rank correlation on city-grouped folds of the training cities (Appendix J.1) for every subspace size k and temperature τ we considered. All runs use one component per class ($M = 1$), standardized coordinates and the two-learner mean. On LOS/NLOS identification ρ barely changes between $k = 32$ and 64 and between $\tau = 0$ and 0.3, while $k = 16$ is lower, so we use $k = 32$ and $\tau = 0.1$ on this plateau. On beam prediction with 32 classes, DiMTE improves with more directions and a softer decision, and we use $k = 64$ and $\tau = 1.0$. The grid was extended to $k = 128$ and $\tau = 1.0$ on the training cities only, where $\tau = 1.0$ lies at the edge of the grid and $k = 128$ ties $k = 64$. DiMTE (learned space) starts from the raw principal coordinates of the same k. Both settings, and afterwards the number of components (Appendix G), were chosen by training-city results only.

The principal directions can come from samples of the training cities alone. DiMTE fits its k principal directions on the samples of the set it scores, which lets target samples shape the space and breaks the independence that Proposition 1 assumes. Directions fitted on samples of the training cities instead give DiMTE a ρ of 0.76 and 0.79 on the Sionna test cities, against 0.78 and 0.79 with the set's own samples. It still leads all 42 and 48 comparators of that test, with every one-sided p below 0.001. On the DeepMIMO test cities it reaches 0.84 and 0.91, against 0.85 and 0.91. So the choice of samples for the projection does not carry the result.

Table 12: Choice of the subspace size k and the temperature τ of DiMTE on the training cities (city-grouped folds) against the measured transfer, scored by per-target Spearman ρ and gap recovered. The hard arg-max is $\tau = 0$, and the coordinates are standardized. Shaded rows use the paper's setting for that task.

		LOS/NLOS		Beam prediction	
k	τ	ρ	gap	ρ	gap
16	0	+0.58	75%	+0.47	62%
16	0.1	+0.58	75%	+0.49	73%
16	0.3	+0.60	77%	+0.51	76%
32	0	+0.63	82%	+0.59	72%
32	0.1	+0.64	82%	+0.60	77%
32	0.3	+0.64	81%	+0.62	82%
64	0	+0.63	86%	+0.62	77%
64	0.1	+0.64	85%	+0.63	79%
64	0.3	+0.64	84%	+0.65	84%
64	1	–	–	+0.67	82%
128	0.3	–	–	+0.64	80%
128	1	–	–	+0.67	84%

A new task can start from the 1-NN rule, before any transfer matrix exists. The settings above need the transfer matrix of the training cities once per task, and Table 13 asks what a new task loses without that step. The settings chosen for the other task lose up to -0.09 in ρ, for LOS/NLOS identification on the Sionna test cities, because the number of principal coordinates matters there. The 1-NN source classifier has only that number as a setting. At $k = 32$ it changes ρ by between -0.03 and +0.03 on the four sets, so one fixed rule on the LOS/NLOS coordinates serves both tasks. The value $k = 32$ is the LOS/NLOS setting of the training cities. The same rule at $k = 64$ loses up to 0.09 on LOS/NLOS identification. On the Sionna test cities the rule at $k = 32$ reaches 0.76 on LOS/NLOS identification, against 0.75 for the strongest comparator. This comparison was run after the method was fixed, and it changes no setting.

Table 13: What a new task loses without its own hyperparameters. DiMTE runs with the settings the training cities chose for the other task ($k = 32$, $\tau = 0.1$ and $M = 32$ for LOS/NLOS identification, and $k = 64$, $\tau = 1$ and $M = 128$ for beam prediction), and with a 1-NN source classifier, which has no setting except the number of principal coordinates, all in the paper's projection. Per-target Spearman ρ against the measured transfer, with the difference from the paper's setting in brackets. Shaded rows use the paper's setting.

	DeepMIMO test cities		Sionna test cities	
Source classifier and settings	LOS/NLOS	Beam	LOS/NLOS	Beam
Settings chosen for the task (the paper's)	0.85	0.91	0.78	0.79
Settings chosen for the other task	0.85 (-0.00)	0.89 (-0.02)	0.69 (-0.09)	0.75 (-0.04)
1-NN rule, $k = 32$	0.88 (+0.03)	0.91 (+0.00)	0.76 (-0.03)	0.80 (+0.01)
1-NN rule, $k = 64$	0.80 (-0.05)	0.91 (+0.00)	0.69 (-0.09)	0.79 (-0.00)

G Choice of the Number of Mixture Components

This appendix chooses the number of components per class M on the training cities and checks how much the choice matters on the test cities.

More components improve DiMTE on all three tasks, up to a few dozen. Table 14 varies M on the training cities, with the subspace size and temperature chosen for $M = 1$. The rank correlation rises from 0.64 to 0.74 on LOS/NLOS identification, from 0.67 to 0.79 on beam prediction, and from

Table 14: Choice of the number of components per class M of the summary on the training cities (city-grouped folds) against the measured transfer. $M = 1$ is the tied Gaussian summary. Per-target Spearman ρ and share of the random-to-genie gap recovered. The subspace size and temperature are those chosen with $M = 1$, in Table 12 for LOS/NLOS identification and beam prediction and in Appendix Q for UL-AoA sector classification. Best in bold, second best underlined, with printed ties marked alike. Shaded cells use the paper's setting for that task, the largest unrounded ρ.

	LOS/NLOS		Beam prediction		UL-AoA sector	
Components per class M	ρ	gap	ρ	gap	ρ	gap
1	0.64	82%	0.67	82%	0.77	82%
4	0.68	84%	0.70	77%	0.83	86%
8	0.71	85%	0.73	71%	0.84	86%
16	0.72	84%	0.77	78%	0.85	88%
32	**0.74**	85%	0.78	82%	0.86	89%
64	**0.74**	85%	**0.79**	80%	**0.88**	**91%**
128	**0.74**	**88%**	**0.79**	**83%**	**0.88**	**91%**

0.77 to 0.88 on UL-AoA sector classification (Appendix Q), and it flattens beyond 32 components. We use the M with the largest training ρ on each task, namely $M = 32$, $M = 128$ and $M = 128$. A beam class holds about 30 samples of a dataset on average. With more components than a class has samples, every sample becomes its own component, and the classifier becomes a kernel classifier on the stored samples, with the shrunk variance of (7) as its bandwidth. The summary then grows from $Ck + k + C$ numbers toward the set of projected embeddings, which is still far smaller than one trained model per source.

On the DeepMIMO test cities the result is robust to M. The test scores came after the choice of M and did not inform it. DiMTE reaches 0.82 to 0.85 on LOS/NLOS identification for every M we scored from 4 to 64, and 0.82 to 0.91 on beam prediction for M from 4 to 128. Every value lies above the strongest comparator of that set (0.72 and 0.77). On the Sionna test cities M matters more for LOS/NLOS identification. There the ranking moves by 0.078 over M, and one or four components fall below the strongest comparator of Table 1 (Figure 6c).

Several components do not help the learned variant on the test cities. For the learned variant, the components are assigned once in the starting coordinates, and the centroids move with the learned projection, so training stays differentiable. On the training cities, eight components raise DiMTE (learned space) from 0.80 to 0.83 on LOS/NLOS identification, while four components leave beam prediction at 0.69 against 0.70. On the DeepMIMO test cities the eight-component variant reaches 0.81, level with one component (Table 46).

H Ablations of DiMTE

This appendix removes the design choices of DiMTE one at a time, to show which of them carry the ranking.

Table 15: Ablations of DiMTE on the training cities (city-grouped folds) against the measured transfer, scored by per-target Spearman ρ and gap recovered. Each variant changes one choice of the paper's setting. The row with a loss other than the transfer metric evaluates $1-$F1 on beam prediction and the classification error on LOS/NLOS identification. Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction	
Variant	ρ	gap	ρ	gap
DiMTE (paper setting)	+0.74	85%	+0.79	83%
hard arg-max instead of softmax	+0.74	84%	+0.75	78%
one component per class ($M = 1$)	+0.64	82%	+0.67	82%
raw principal coordinates (no standardization)	+0.60	74%	+0.69	81%
no class conditioning (label-free Gaussian)	+0.22	69%	+0.36	54%
one component per class, untied class variances	+0.04	34%	+0.60	58%
loss other than the transfer metric	+0.75	85%	+0.67	67%

Table 15 changes one choice of DiMTE at a time, on training-city folds against the two-learner mean.

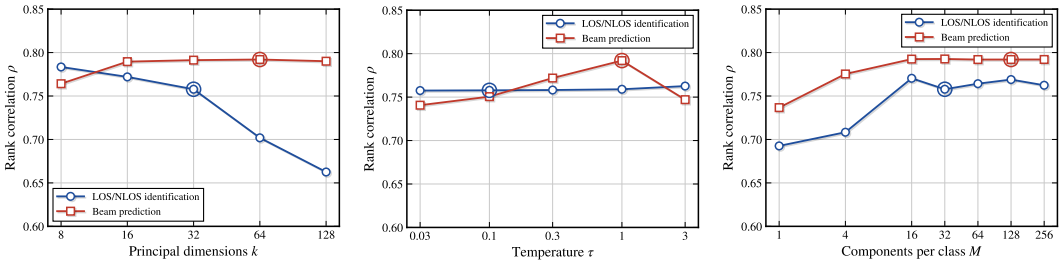
Class conditioning and three choices inside the summary carry DiMTE, namely standardized coordinates, several components per class and a tied variance. Standardized principal coordinates raise ρ on LOS/NLOS identification from 0.60 to 0.74. Without standardization the leading direction dominates the classifier, and the variance floor suppresses the other directions. One component per class lowers ρ to 0.64, and untying the variances of that one-component summary collapses it to 0.04. Untied variances turn the linear decision boundaries into quadratic ones. Those boundaries bend with each source's per-class spread, which is itself noisy from a few hundred samples. They also extrapolate poorly to samples of another site, where the classifier is evaluated. Class conditioning matters as well. Without it DiMTE falls to 0.22, the level of the label-free baselines.

Two choices do not matter on LOS/NLOS identification. The hard arg-max reaches 0.74 against the softmax's 0.74, and the classification error as the readout, instead of the weighted F1, gives 0.75, because the two readouts nearly coincide on a binary task. On beam prediction with 32 classes the picture shifts. Standardization matters as much (0.69 against 0.79), and so do several components (0.67 with one) and class conditioning (0.36). Untied one-component variances cost less there (0.60), because 32 class means already constrain the classifier. The decision temperature matters more here (0.75 for the hard arg-max), which is why the beam setting uses $\tau = 1$. The $1-$F1 readout, in place of the L1-RSRP difference, gives 0.67 and lowers the gap recovered from 83% to 67%. The metric readout is therefore what makes the selected source good in dB.

A check on the test sets, run after the method was fixed, shows the same and changes no setting. The beam summaries rank the sources of the Sionna test cities at 0.63 instead of 0.79 under the $1-$F1 readout, below the strongest Sionna comparator (0.68). On the DeepMIMO test cities they reach 0.80 instead of 0.91, still above the strongest DeepMIMO comparator (0.77). The metric readout leads the $1-$F1 readout by +0.16 and +0.10, computed from unrounded values, with 95% city-clustered intervals [+0.13, +0.18] and [+0.06, +0.15] and one-sided p below 0.001 and below 0.001. The $1-$F1 readout needs only the labels, so without the target's beam-power vector DiMTE falls below the strongest comparator on the Sionna test cities.

Table 16: DiMTE with one choice changed at a time, on both test sets, all else at the paper's setting. Per-target Spearman ρ. *Tied full* replaces the tied diagonal variance by a tied full covariance with Ledoit-Wolf shrinkage, and *class-specific diagonal* gives every class its own shrunk diagonal variance. The prior weight is β in (4). These variants were run after the method was fixed and are reported, not adopted. Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction	
Variant	DeepMIMO test	Sionna test	DeepMIMO test	Sionna test
Tied diagonal variance ($\beta = 1$, softmax)	+0.85	+0.78	+0.91	+0.79
Tied full covariance	+0.86	+0.79	+0.91	+0.81
Class-specific diagonal variances	+0.53	+0.08	+0.80	+0.58
Prior weight $\beta = 0$	+0.82	+0.78	+0.89	+0.73
Prior weight $\beta = 0.5$	+0.84	+0.78	+0.90	+0.78
Prior weight $\beta = 2$	+0.85	+0.78	+0.90	+0.77
Hard arg-max ($\tau \to 0$)	+0.85	+0.78	+0.90	+0.74



(a) Principal dimensions (b) Temperature (c) Components per class

Figure 6: How much each setting of the summary changes the ranking on the Sionna test cities. One setting varies at a time around the chosen value, which each panel circles. The projection is fitted on samples of the training cities, so no test sample enters the space at any point of the sweep. The sweep was run after the settings were fixed and changes none of them. The two tasks respond differently. The number of principal dimensions moves ρ by 0.121 on LOS/NLOS identification and by 0.028 on beam prediction, while the temperature moves it by 0.051 on beam prediction over a hundredfold span and by 0.005 on LOS/NLOS identification. The settings chosen on the training cities are near-optimal for beam prediction on all three axes. On LOS/NLOS identification a smaller k would have scored higher.

I OTHER ENCODERS

This appendix asks whether DiMTE needs the LWM encoder. It replaces LWM with a task-trained CNN and with a public wireless foundation model.

A task-trained encoder does not replace the pretrained one in DiMTE. We train the architecture of the raw-CSI CNN on the labeled samples of all training datasets together and evaluate its penultimate features at the paper's setting. The received power spans tens of dB across the pooled sites, which left the batch-normalization statistics of the raw input unusable. Each sample therefore enters at unit RMS, with its log power relative to its dataset's median as a fourth input channel. The CNN reaches a validation weighted F1 of 0.87 on LOS/NLOS identification and 0.67 on beam prediction. Its features nevertheless rank Sionna test sources at 0.15 and 0.54, against 0.78 and 0.79 with LWM, and DeepMIMO test sources at 0.50 and 0.69 (Table 17). LWM itself is close to the self-supervised encoder one might train as a fallback, since its masked channel modeling read unlabeled channels of 20 of our training cities and of no test city (Appendix C). The measured transfer does not rest on LWM alone. The two-learner mean averages fine-tuned LWM and a CNN that shares no features with it, and Table 24 scores every method against each deployment model alone.

On a public encoder that is not a backbone of the deployment models, DiMTE leads every comparator with the same access on three of the four test sets and ties the best on the fourth. WavesFM v1.0 is the released multimodal wireless foundation model of Aboulfotouh & Abou-Zeid (2025), and its vision branch reads image-like inputs such as channels. Its authors pretrained it by masked reconstruction on spectrograms and in-phase and quadrature (IQ) recordings, and we use the released pretrained-only checkpoint frozen, so no label of our tasks has reached it. Each channel goes through the model's own preprocessing for DeepMIMO channels. The real and imaginary parts form two image channels, which are resized bicubically to 224×224 and standardized with fixed statistics of the training cities, so the input keeps the received power. The ViT encoder has 8 blocks of width 256 and 16×16 patches, and every embedding is scaled to unit length, as the LWM embeddings are. The training cities choose the embedding token, the class token or the mean patch token, together with k, τ and M in the two steps of Appendices F and G. They pick the mean patch token with $k = 32$, $\tau = 0.1$ and $M = 32$ for LOS/NLOS identification and the mean patch token with $k = 128$, $\tau = 1$ and $M = 128$ for beam prediction. A comparator with the same access reads no encoder or reads the same WavesFM features. We therefore recompute LEEP and NCE on both source proxies and the joint KL on WavesFM features, on both tokens and in both projections. The best of these and of the encoder-free comparators on each test set is the comparator we report, a choice that favors the comparators.

On the Sionna test cities DiMTE reaches 0.74 on LOS/NLOS identification and 0.77 on beam prediction, against 0.72 and 0.68 for the best comparator with the same access (largest one-sided p <0.001 and <0.001). On the DeepMIMO test cities it reaches 0.80 against 0.68 on LOS/NLOS identification (p <0.001), and 0.78 against 0.75 for the directed joint KL on WavesFM on beam prediction, a tie at p 0.231 (Table 17). Comparators on LWM embeddings read an encoder that DiMTE does not use here. Against them as well, DiMTE leads on two test sets and is level on the other two. The two level cases are Sionna LOS/NLOS identification, at 0.74 against 0.75 ($p = 0.711$), and DeepMIMO beam prediction, at 0.78 against 0.77 ($p = 0.373$). On LWM embeddings DiMTE reaches 0.78 and 0.79 on the Sionna test cities and 0.85 and 0.91 on the DeepMIMO test cities. The encoder therefore moves the correlation, most on DeepMIMO beam prediction, while the construction keeps its lead over comparators with the same access without LWM. ContraWiMAE (Guler et al., 2026) also accepts our channels, but its checkpoints are not released, and WiFo (Liu et al., 2025) needs a time axis that our channels lack.

Table 17: DiMTE on other encoders. *CNN features* are the penultimate features of a CNN trained on the labeled samples of all training datasets together (the raw-CSI CNN architecture of the two-learner mean, with each sample at unit RMS and its log power as a fourth input channel), evaluated at the paper's setting in place of the LWM embedding. Entries are per-target Spearman ρ. The margin rows give the paired margin of the LWM mixture over the CNN-feature mixture and its 95% interval. *WavesFM features* come from the released pretrained-only checkpoint of a public wireless foundation model (Aboulfotouh & Abou-Zeid, 2025), frozen and fed through its own channel preprocessing. Its embedding token (class or mean patch token), k, τ and M are chosen on the training cities as ours are. *Same access* is the strongest comparator that reads no encoder or reads WavesFM. For it, LEEP and NCE on both source proxies and the joint KL are recomputed on WavesFM features, on both tokens and in both projections. *Any representation* adds every comparator on LWM embeddings. Shaded rows use the paper's setting.

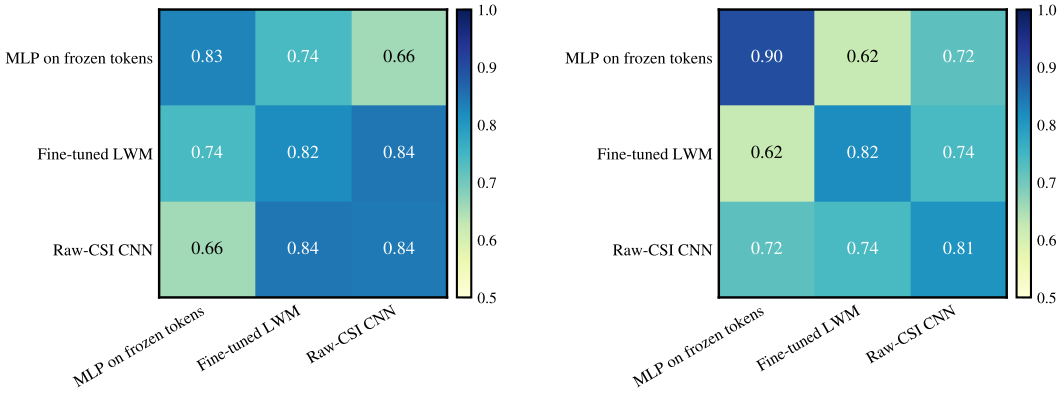
	LOS/NLOS		Beam prediction	
Estimate	DeepMIMO test	Sionna test	DeepMIMO test	Sionna test
Mixture classifier on LWM (DiMTE)	+0.85	+0.78	+0.91	+0.79
Mixture classifier on CNN features	+0.50	+0.15	+0.69	+0.54
k-NN classifier on CNN features	+0.49	+0.16	+0.68	+0.52
Parzen classifier on CNN features	+0.49	+0.16	+0.68	+0.52
Margin of LWM over CNN features	+0.35	+0.63	+0.22	+0.25
its 95% interval	[+0.32, +0.39]	[+0.55, +0.71]	[+0.17, +0.27]	[+0.22, +0.28]
DiMTE on WavesFM features	+0.80	+0.74	+0.78	+0.77
Best comparator, same access	+0.68	+0.72	+0.75	+0.68
Best comparator, any representation	+0.72	+0.75	+0.77	+0.68

J Results on the Training Cities

This appendix holds three results: (i) Table 18 and Figure 7, the four deployment models of Section 5.2 in metric units and their agreement, (ii) Table 19, the training-set comparison that Appendix J.1 summarizes, and (iii) Figure 8, the estimate of DiMTE against the measured transfer on the DeepMIMO test set.

Table 18: The transfer matrix depends on the deployment model. For each deployment model on the training cities, the table gives four quantities: (i) the in-site metric (a model tested on its own site), (ii) the cross-site metric, (iii) the per-target agreement between one training seed and the mean of the others (reliability), and (iv) the per-target agreement with fine-tuned LWM, or with the CNN for the fine-tuned LWM row itself. Metrics are 1−F1 (LOS/NLOS identification) and the L1-RSRP difference in dB (beam prediction, 8-antenna input).

	LOS/NLOS				Beam prediction			
Deployment model	in-site	cross-site	reliability ρ	agreement ρ	in-site	cross-site	reliability ρ	agreement ρ
Linear on frozen [CLS]	0.025	0.083	0.81	0.70	–	–	–	–
MLP on frozen tokens	0.021	0.089	0.83	0.74	9.7	19.1	0.90	0.62
Fine-tuned LWM	0.015	0.110	0.82	0.84	3.9	12.3	0.82	0.74
Raw-CSI CNN	0.010	0.138	0.84	0.84	6.6	14.3	0.81	0.74



(a) LOS/NLOS identification

(b) Beam prediction

Figure 7: How much the deployment model decides the transfer matrix. Off the diagonal, the mean per-target rank correlation between the matrices of two deployment models on the training cities. On the diagonal, one training seed of a deployment model against the mean of its other seeds. Fine-tuned LWM and the raw-CSI CNN agree at 0.84 and 0.74, and the MLP on frozen tokens agrees with the two of them at 0.70 and 0.67 on average.

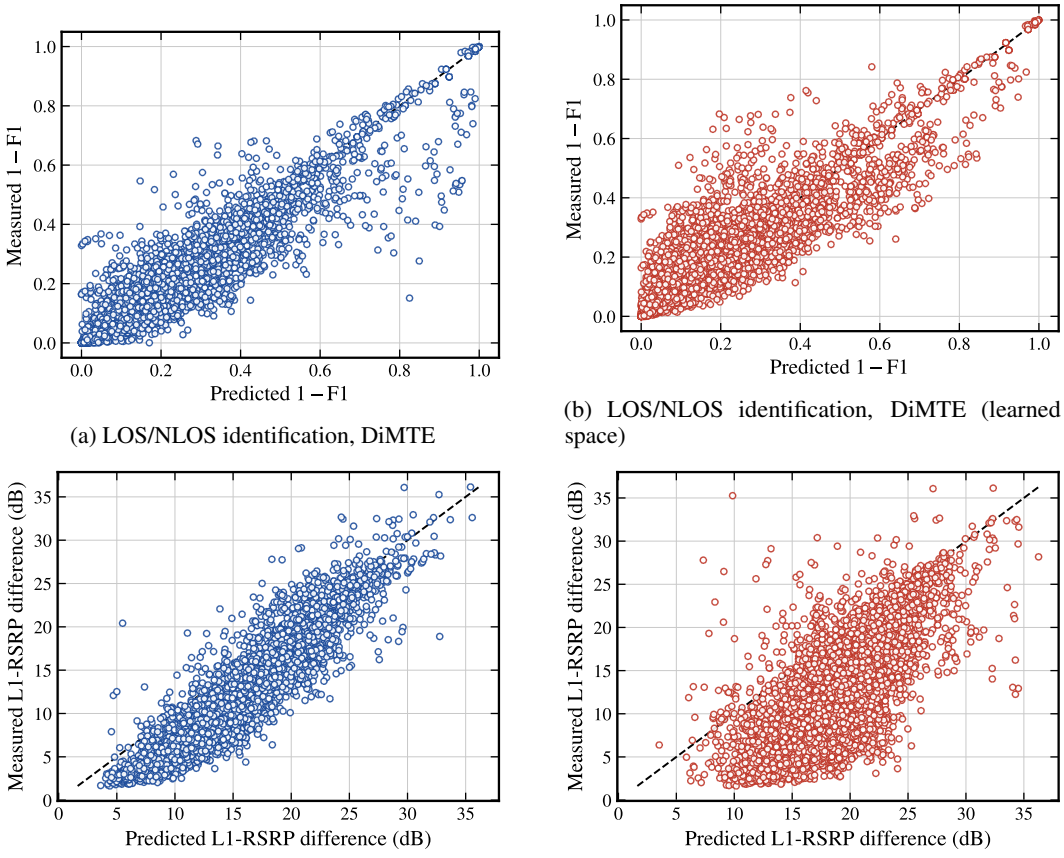
J.1 Alignment on the Training Cities

On the training cities the methods that learn from the transfer matrix lead on one task each, and neither lead holds on the test cities. Table 19 in Appendix J reports every method run on the training cities, under city-grouped folds, scored against the two-learner mean. First, DiMTE needs standardized principal coordinates and several components per class to track the deployment models. With raw coordinates it reaches $\rho = 0.60$ on LOS/NLOS identification, and with one component 0.64. With both choices it reaches 0.74, the best result in Table 19 among methods that do not observe transfer outcomes. The hyperparameters of DiMTE were chosen by this score, so its rows in Table 19 are the best of a grid. The learned methods, by contrast, are scored on folds they were not trained on. Only other source classifiers evaluated the same way do slightly better (Table 29). Second, DiMTE (learned space) learns a source ranking closer to the measured transfer on LOS/NLOS identification. There it reaches 0.80, exceeding the best baseline (Pairwise transfer predictor, LWM descriptors) by +0.11 (p below 0.001). On beam prediction a baseline leads on the training cities. The best baseline there (Pairwise transfer predictor, LWM descriptors) reaches 0.89, ahead of DiMTE at 0.79 and DiMTE (learned space) at 0.70. That baseline also learns from the transfer matrix of the training

cities. On the DeepMIMO test cities it falls to 0.64, against 0.91 for DiMTE (Table 20), and Table 23 shows the same on the Sionna test cities. The training lead of DiMTE (learned space) on LOS/NLOS identification does not persist there either.

Standardized coordinates, several components and tied variances are the choices that matter. Appendix H removes the design choices of DiMTE one at a time. Without standardization DiMTE reaches 0.60, with one component 0.64, and with one component and untied variances 0.04, against 0.74 with all choices. On the binary task, the decision temperature and the readout have much smaller effects.

The estimate of DiMTE tracks the transfer metric in its own units, with an offset. Figure 8 plots every cross-city pair of the DeepMIMO test cities. On LOS/NLOS identification DiMTE predicts the $1-\mathrm{F1}$ of the transferred model with a Pearson correlation of 0.93 and a root-mean-square error of 0.070, and DiMTE (learned space) with 0.91 and 0.076. On beam prediction DiMTE reaches 0.93 and 4.2 dB, against a cross-site mean of 10.6 dB, and DiMTE (learned space) 0.75 and 8.4 dB. DiMTE overestimates the smallest L1-RSRP differences by a few dB. The classifier of the summary is weaker than the fine-tuned and convolutional models it stands in for, which is the mismatch term $\Delta_{s,t}$ of Proposition 1. The conformal intervals of Appendix K.1 fit the mapping from estimate to metric and absorb such an offset. **The target samples that DiMTE reads do not lift its correlation.**



(a) LOS/NLOS identification, DiMTE

(b) LOS/NLOS identification, DiMTE (learned space)

(c) Beam prediction, DiMTE

(d) Beam prediction, DiMTE (learned space)

Figure 8: Predicted against measured transfer metric for every cross-city source-target pair of the DeepMIMO test set, against the two-learner mean. The estimate is in the units of the deployment metric, so the dashed identity line is the perfectly calibrated predictor.

DiMTE reads each target's seed-0 training split, while seed j of the transfer matrix tests on that target's seed-j test split. The seed-0 column is therefore scored on samples disjoint from the ones that DiMTE reads, and the later seeds redraw the split. On the DeepMIMO test set DiMTE reaches 0.84 and 0.90 against seed 0 alone, 0.84 and 0.89 against the later seeds, and 0.85 and 0.91 against the mean of the three that the paper uses. The samples it reads therefore do not help it.

Table 19: Source ranking against the measured transfer on the training cities (city-grouped folds). Per-target Spearman correlation ρ between score and measured transfer, and share of the random-to-genie gap recovered by the selected source. Rows are grouped by what each method uses. A dash marks a ρ that is undefined because the method gives every source the same score for some target, which happens for CKA on the two class centroids of LOS/NLOS identification. Best in bold, second best underlined. Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction (8-antenna input)	
Distance	ρ	gap	ρ	gap
Class prior				
Class prior (total variation)	+0.20	49%	+0.65	61%
Class prior (KL, directed)	+0.21	49%	+0.69	57%
Raw channels				
Label-free SW	+0.24	51%	+0.62	66%
Label-aware centroid	+0.14	46%	+0.38	24%
Label-aware SW	+0.26	59%	+0.44	42%
Joint KL (directed), raw coordinates	+0.15	57%	+0.62	53%
DiMTE, one component	+0.20	32%	+0.51	49%
OTDD	+0.26	56%	+0.64	67%
UMAP, 2-d	+0.26	32%	+0.29	19%
UMAP, 16-d	+0.29	49%	+0.31	12%
UMAP, 2-d, label-supervised	+0.29	28%	+0.18	−13%
Task2Vec	+0.14	28%	+0.63	62%
LWM embeddings, no source model				
Label-aware centroid	+0.29	61%	+0.47	42%
Label-aware SW	+0.31	68%	+0.43	33%
Joint KL (directed), LWM coordinates	+0.30	78%	+0.69	79%
MMD, label-free	+0.17	46%	+0.28	31%
MMD, label-aware	+0.31	56%	+0.46	37%
CKA on class centroids	–	−72%	+0.10	17%
Task2Vec	+0.18	43%	+0.58	56%
Transferability estimators				
LEEP, closed-form source	+0.41	68%	−0.05	−23%
NCE, closed-form source	+0.36	54%	+0.34	39%
OTCE, conditional entropy	+0.24	57%	+0.50	48%
OTCE, transport + entropy	+0.19	61%	+0.49	51%
LEEP, closed-form source, standardized	+0.59	82%	+0.37	17%
NCE, closed-form source, standardized	+0.59	78%	+0.47	39%
LEEP on our mixture classifier	+0.68	<u>86%</u>	+0.45	44%
NCE on our mixture classifier	+0.68	84%	+0.52	45%
OTCE, conditional entropy, standardized	+0.21	57%	+0.52	48%
OTCE, transport + entropy, standardized	+0.22	72%	+0.56	59%
LEEP, model trained per source	+0.34	83%	+0.35	26%
NCE, model trained per source	+0.44	42%	+0.46	46%
N-LEEP, model trained per source	+0.23	28%	+0.39	36%
LogME, model trained per source	+0.07	16%	+0.25	5%
H-score (regularized), model per source	+0.13	13%	+0.35	17%
GBC, model trained per source	+0.23	21%	+0.26	9%
TransRate, model trained per source	+0.01	24%	+0.44	33%
PARC, model trained per source	+0.31	49%	+0.29	43%
Learned without transfer supervision				
Self-supervised Deep Sets, LWM embeddings	+0.10	62%	+0.34	66%
Learned from the library transfer matrix				
Pairwise transfer predictor, raw descriptors	+0.62	74%	+0.81	75%
Pairwise transfer predictor, LWM descriptors	+0.69	80%	**+0.89**	81%
Directed Deep Sets, raw channels	+0.65	74%	+0.67	65%
Directed Deep Sets, LWM embeddings	+0.58	75%	<u>+0.82</u>	78%
Directed Set Transformer, LWM embeddings	+0.48	71%	+0.61	57%
DiMTE (ours)				
Mixture classifier (default)	<u>+0.74</u>	85%	+0.79	**83%**
DiMTE (learned space)	**+0.80**	**88%**	+0.70	<u>82%</u>

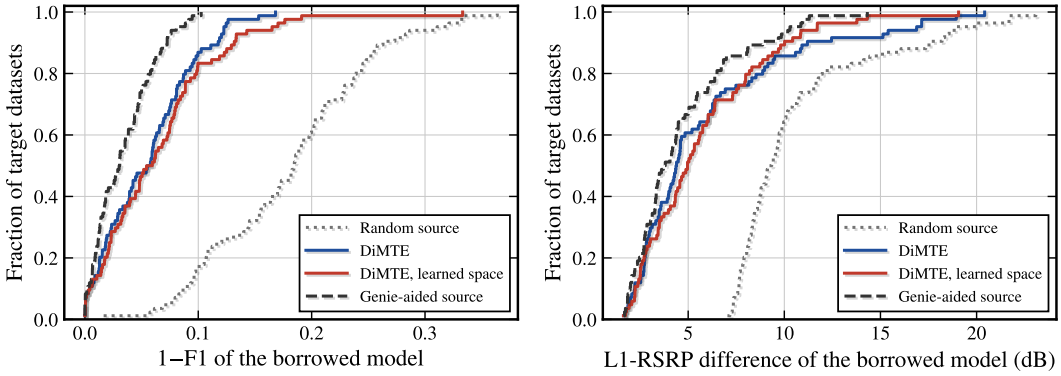
Table 20: Source selection on the DeepMIMO test cities, against the mean transfer of fine-tuned LWM and a raw-CSI CNN. Per-target Spearman ρ, written $0.78(1)$ where the bracketed digit is the standard error of its city means in the last place, the share of the gap between a random and the genie-aided source that the selection closes, and in square brackets the selected source's $1-$F1 or its L1-RSRP difference in dB, with Top-1/1 accuracy. The rows under *DiMTE with each source classifier* grade different classifiers induced by the source in the same way, and the shaded mixture is the default, fixed before the Sionna test cities existed. Best in bold and second best underlined over every row, DiMTE rows included. Shaded rows use the paper's setting. The row marked *validated settings* selects the learned variant's start and number of components on the training cities (Appendix N).

	LOS/NLOS		Beam prediction (8-antenna input)	
Distance	ρ	gap [metric]	ρ	gap [metric]
Class prior				
Class prior (total variation)	+0.35(2)	32% [0.133]	+0.71(2)	71% [6.3, 52%]
Class prior (KL, directed)	+0.36(2)	33% [0.131]	+0.73(2)	70% [6.4, 53%]
Raw channels				
Label-free SW	+0.20(1)	37% [0.126]	+0.70(2)	69% [6.5, 52%]
Label-aware centroid	+0.13(2)	−39% [0.238]	+0.61(2)	43% [8.1, 46%]
Label-aware SW	+0.19(2)	−61% [0.270]	+0.57(2)	3% [10.4, 40%]
Joint KL (directed), raw coordinates	+0.17(2)	54% [0.100]	+0.69(2)	69% [6.5, 53%]
DiMTE, one component	+0.29(2)	43% [0.117]	+0.63(2)	53% [7.4, 50%]
OTDD	+0.22(1)	29% [0.138]	+0.72(1)	65% [6.7, 50%]
UMAP, 2-d	+0.17(1)	23% [0.147]	+0.48(2)	9% [10.1, 42%]
UMAP, 16-d	+0.17(1)	−15% [0.203]	+0.50(2)	5% [10.3, 40%]
UMAP, 2-d, label-supervised	+0.23(1)	33% [0.132]	+0.43(2)	−15% [11.5, 37%]
Task2Vec	+0.01(2)	26% [0.142]	+0.73(2)	57% [7.2, 52%]
LWM embeddings, no source model				
Label-aware centroid	+0.39(3)	29% [0.138]	+0.60(2)	62% [6.9, 50%]
Label-aware SW	+0.40(3)	45% [0.114]	+0.53(2)	50% [7.6, 49%]
Joint KL (directed), LWM coordinates	+0.48(2)	80% [0.063]	+0.77(2)	78% [6.0, 53%]
MMD, label-free	+0.29(2)	55% [0.100]	+0.32(3)	34% [8.6, 44%]
MMD, label-aware	+0.38(3)	42% [0.119]	+0.58(2)	52% [7.5, 48%]
CKA on class centroids	+0.19(2)	5% [0.173]	+0.18(2)	27% [9.0, 43%]
Task2Vec	+0.23(3)	36% [0.127]	+0.68(2)	50% [7.6, 50%]
Transferability estimators				
LEEP, closed-form source	+0.57(3)	66% [0.083]	−0.33(2)	−99% [16.5, 16%]
NCE, closed-form source	+0.54(3)	51% [0.105]	+0.50(2)	50% [7.6, 51%]
OTCE, conditional entropy	+0.32(1)	26% [0.142]	+0.65(2)	51% [7.6, 51%]
OTCE, transport + entropy	+0.29(2)	73% [0.073]	+0.60(3)	71% [6.4, 52%]
LEEP, closed-form source, standardized	+0.65(3)	<u>86%</u> [0.054]	+0.45(2)	29% [8.9, 39%]
NCE, closed-form source, standardized	+0.65(3)	70% [0.077]	+0.59(3)	50% [7.6, 49%]
LEEP on our mixture classifier	+0.70(4)	84% [0.057]	+0.72(3)	56% [7.3, 53%]
NCE on our mixture classifier	+0.72(2)	78% [0.066]	+0.74(3)	63% [6.9, 52%]
LEEP, closed-form source, our projection	+0.64(3)	77% [0.068]	+0.47(2)	35% [8.5, 39%]
NCE, closed-form source, our projection	+0.63(3)	72% [0.074]	+0.61(2)	53% [7.4, 48%]
LEEP on our mixture, our projection	+0.70(4)	84% [0.057]	+0.72(3)	57% [7.2, 52%]
NCE on our mixture, our projection	+0.72(2)	82% [0.060]	+0.74(3)	59% [7.1, 53%]
OTCE, conditional entropy, standardized	+0.39(2)	64% [0.087]	+0.69(2)	55% [7.3, 52%]
OTCE, transport + entropy, standardized	+0.35(2)	75% [0.070]	+0.61(3)	62% [6.9, 52%]
JC-NCE, joint transport	+0.33(1)	58% [0.096]	+0.54(2)	43% [8.1, 46%]
WDJE, bound on our mixture	+0.25(2)	28% [0.139]	+0.67(2)	59% [7.1, 52%]
LEEP, model trained per source	+0.56(3)	81% [0.062]	+0.60(2)	46% [7.8, 49%]
NCE, model trained per source	+0.60(3)	68% [0.080]	+0.65(3)	50% [7.6, 50%]
N-LEEP, model trained per source	+0.36(2)	−40% [0.239]	+0.57(2)	51% [7.6, 48%]
LogME, model trained per source	+0.19(2)	15% [0.158]	+0.43(3)	34% [8.6, 45%]
H-score (regularized), model per source	+0.24(3)	12% [0.163]	+0.49(3)	38% [8.4, 46%]
GBC, model trained per source	+0.37(3)	34% [0.131]	+0.50(3)	51% [7.6, 47%]
TransRate, model trained per source	+0.01(3)	−14% [0.201]	+0.58(2)	46% [7.9, 47%]
PARC, model trained per source	+0.52(2)	62% [0.090]	+0.56(2)	53% [7.5, 48%]
Learned without transfer supervision				
Self-supervised Deep Sets, LWM	+0.27(2)	61% [0.090]	+0.39(1)	67% [6.6, 50%]
Learned from the library transfer matrix				
Pairwise transfer predictor, raw descriptors	+0.24(2)	36% [0.127]	+0.58(1)	67% [6.6, 49%]
Pairwise transfer predictor, LWM descriptors	+0.51(2)	61% [0.091]	+0.64(2)	62% [6.9, 50%]

Distance	LOS/NLOS ρ	LOS/NLOS gap [metric]	Beam prediction (8-antenna input) ρ	Beam prediction (8-antenna input) gap [metric]
Directed Deep Sets, raw channels	+0.17(2)	−69% [0.281]	+0.22(1)	−38% [12.9, 25%]
Directed Deep Sets, LWM embeddings	+0.51(4)	72% [0.074]	+0.69(2)	61% [6.9, 49%]
Directed Set Transformer, LWM embeddings	+0.42(4)	−101% [0.328]	+0.46(1)	17% [9.6, 38%]
Baselines that read the target's L1-RSRP sweep				
Directed joint KL, RSRP (1 dB)	–	–	+0.74(2)	73% [6.2, 52%]
Directed joint KL, RSRP (3 dB)	–	–	+0.69(2)	71% [6.4, 51%]
Directed joint KL, RSRP (10 dB)	–	–	+0.50(3)	70% [6.4, 51%]
Class-prior KL, RSRP	–	–	+0.74(2)	70% [6.5, 52%]
Label-aware centroid, RSRP	–	–	+0.61(3)	60% [7.0, 49%]
DiMTE with each source classifier (ours)				
Mixture classifier (default)	+0.85(1)	85% [0.055]	**+0.91**(1)	79% [5.9, 54%]
k-NN classifier	**+0.87**(1)	**89%** [0.050]	**+0.91**(1)	81% [5.8, 54%]
Parzen (kernel) classifier	**+0.87**(1)	**89%** [0.050]	+0.90(1)	**83%** [5.7, 54%]
One mean per class	+0.60(2)	82% [0.060]	+0.79(1)	78% [5.9, 53%]
DiMTE (learned space)	+0.80(1)	79% [0.064]	+0.76(1)	**83%** [5.7, 54%]
DiMTE (learned space), validated settings	+0.81(1)	59% [0.094]	**+0.91**(1)	78% [5.9, 55%]
Random source	–	0% [0.180]	–	0% [10.6, 37%]
Genie-aided source	–	100% [0.034]	–	100% [4.6, 56%]
One ordering for every target	+0.58	–	+0.72	–

K Detailed Wireless Results

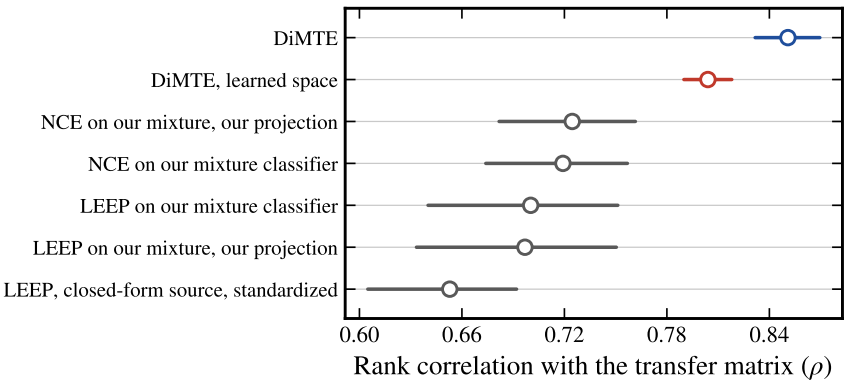
This appendix holds the tables and figures of Section 6.1 and Appendix K.1: (i) every method on the Sionna test cities, all the methods behind Table 1, (ii) the rank correlations and paired margins of DiMTE and its learned variant against the baselines on the DeepMIMO test cities, (iii) the metric of the selected source across cities, (iv) directedness on actual data, (v) the source classifiers inside DiMTE and the bootstrap that also resamples the candidate sources, (vi) the calibrated intervals, and (vii) the two application curves.



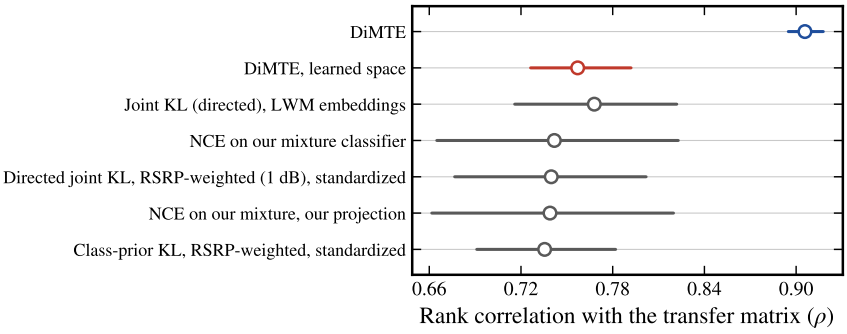
(a) LOS/NLOS identification

(b) Beam prediction from an eight-antenna sub-array

Figure 9: What an operator gets, across the 59 DeepMIMO test cities. Each curve is the distribution over target datasets of the transfer metric of the source that one rule selects. DiMTE and its learned variant sit close to the genie-aided source and far from a random one.

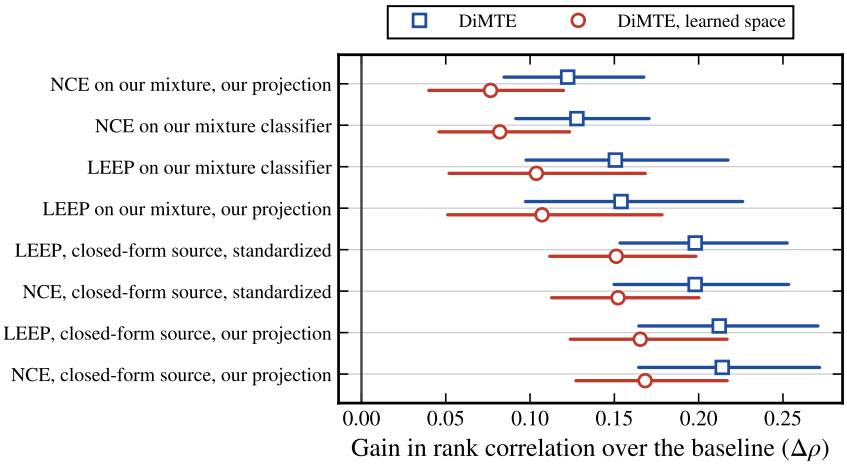


(a) LOS/NLOS identification

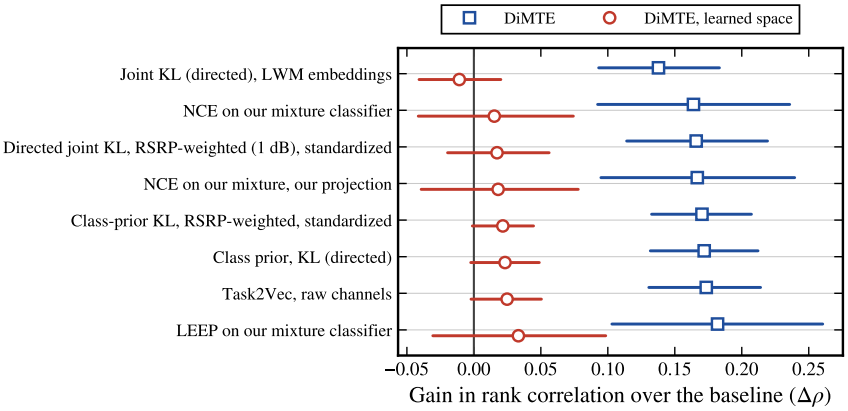


(b) Beam prediction from an eight-antenna sub-array

Figure 10: Source ranking on the 59 DeepMIMO test cities. Each panel shows DiMTE and its learned variant, in color, and the five strongest of the 49 LOS/NLOS and 54 beam baselines, ranked by the mean per-target rank correlation between score and measured transfer. Bars are 95% bootstrap intervals over cities. Figure 11 shows the paired margins on which the claims rest.



(a) LOS/NLOS identification



(b) Beam prediction from an eight-antenna sub-array

Figure 11: On the DeepMIMO test cities, DiMTE ranks sources above every evaluated baseline in mean correlation, and DiMTE (learned space) does so on LOS/NLOS identification. Each row is one baseline, ordered by its own correlation, and each marker is the gain of DiMTE or its learned variant over it with a 95% paired bootstrap interval over cities. The p-values are the one-sided city-clustered permutation tests of Appendix K, so an interval that touches zero can still separate the methods.

Table 21: Source selection on the DeepMIMO test cities, against the mean transfer of fine-tuned LWM and a raw-CSI CNN. Per-target Spearman ρ, written $0.78(1)$ where the bracketed digit is the standard error of its city means in the last place, the share of the gap between a random and the genie-aided source that the selection closes, and in square brackets the selected source's $1-$F1 or its L1-RSRP difference in dB. The rows under *DiMTE with each source classifier* grade different classifiers induced by the source in the same way, and the shaded mixture is the default, fixed before the Sionna test cities existed. Best in bold and second best underlined over every row, DiMTE rows included. Shaded rows use the paper's setting. The last two columns repeat the ranking on the 84 datasets of the Sionna test cities (Table 1 scores every method there). Each group shows its best baseline, plus each task's best by ρ and by gap, and Table 20 lists all methods, including the variant with a learned embedding space of Appendix N.

	LOS/NLOS		Beam prediction		Sionna test, ρ	
Distance	ρ	gap [metric]	ρ	gap [metric]	LOS/NLOS	Beam
Class prior						
Class prior (KL, directed)	+0.36(2)	33% [0.131]	+0.73(2)	70% [6.4]	+0.16(4)	+0.65(1)
Raw channels						
Joint KL (directed), raw coordinates	+0.17(2)	54% [0.100]	+0.69(2)	69% [6.5]	+0.33(1)	+0.68(1)
OTDD	+0.22(1)	29% [0.138]	+0.72(1)	65% [6.7]	+0.34(1)	+0.67(1)
LWM embeddings, no source model						
Joint KL (directed), LWM coordinates	+0.48(2)	80% [0.063]	+0.77(2)	78% [6.0]	+0.40(3)	+0.62(2)
Transferability estimators						
LEEP on our mixture, our projection	+0.70(4)	84% [0.057]	+0.72(3)	57% [7.2]	+0.75(2)	+0.51(3)
NCE on our mixture, our projection	+0.72(2)	82% [0.060]	+0.74(3)	59% [7.1]	+0.75(2)	+0.56(2)
Learned without transfer supervision						
Self-supervised Deep Sets, LWM	+0.27(2)	61% [0.090]	+0.39(1)	67% [6.6]	+0.38(3)	+0.37(2)
Learned from the library transfer matrix						
Directed Deep Sets, LWM embeddings	+0.51(4)	72% [0.074]	+0.69(2)	61% [6.9]	+0.58(3)	+0.44(1)
Baselines that read the target's L1-RSRP sweep						
Directed joint KL, RSRP (1 dB)	–	–	+0.74(2)	73% [6.2]	–	+0.52(2)
DiMTE with each source classifier (ours)						
Mixture classifier (default)	+0.85(1)	85% [0.055]	**+0.91**(1)	79% [5.9]	**+0.78**(1)	+0.79(1)
k-NN classifier	**+0.87**(1)	**89%** [0.050]	**+0.91**(1)	81% [5.8]	+0.76(1)	+0.79(1)
Parzen (kernel) classifier	**+0.87**(1)	**89%** [0.050]	+0.90(1)	**83%** [5.7]	+0.76(1)	**+0.82**(1)
Random source	–	0% [0.180]	–	0% [10.6]	–	–
Genie-aided source	–	100% [0.034]	–	100% [4.6]	–	–
One ordering for every target	+0.58	–	+0.72	–	+0.67	+0.65

Table 22: Rank correlation on the DeepMIMO test cities against each deployment model alone and against their mean, the measured transfer of the paper. Fine-tuned LWM starts from the encoder whose features DiMTE uses, so the CNN column is the measured transfer that is independent of them. The six strongest baselines by the mean are shown, and the best baseline quoted in the text is taken over all of them. Best in bold, second best underlined. Shaded rows use the paper's setting.

	LOS/NLOS			Beam prediction		
Distance	Raw-CSI CNN	Fine-tuned LWM	mean	Raw-CSI CNN	Fine-tuned LWM	mean
DiMTE	**0.85**	**0.79**	**0.85**	**0.91**	**0.86**	**0.91**
DiMTE (learned space)	0.76	**0.79**	0.80	0.76	0.73	0.76
Directed joint KL, RSRP (1 dB)	–	–	–	0.73	0.72	0.74
Class-prior KL, RSRP	–	–	–	0.72	0.73	0.74
NCE on our mixture classifier	0.71	0.69	0.72	0.75	0.71	0.74
LEEP on our mixture classifier	0.69	0.67	0.70	0.73	0.69	0.72
Directed joint KL, RSRP (3 dB)	–	–	–	0.68	0.67	0.69
NCE, model trained per source	0.55	0.65	0.60	0.63	0.66	0.65

	LOS/NLOS		Beam prediction (8-antenna input)	
Distance	ρ	gap [metric]	ρ	gap [metric]

Table 23: Every method on the Sionna test cities, all the methods behind Table 1. The cities lie outside DeepMIMO and are ray traced with Sionna RT (Appendix D), and every method is scored against the mean transfer of fine-tuned LWM and a raw-CSI CNN. Every method ran once with the settings of the DeepMIMO test cities. LEEP, NCE and OTCE on a closed-form source, Task2Vec and the learned encoders fit their principal directions on the training cities, while DiMTE and the other training-free scores fit them on the samples of the set they score. Rows marked *our projection* give LEEP and NCE that projection too. Per-target Spearman ρ, written $0.78(1)$ where the bracketed digit is the standard error of its city means in the last place, the share of the random-to-genie gap recovered, and in square brackets the selected source's mean $1-$F1 or L1-RSRP difference in dB. Best in bold, second best underlined. Shaded rows use the paper's setting. The paired margin of DiMTE over the best baseline is +0.03 [+0.01, +0.05] on LOS/NLOS identification and +0.11 [+0.09, +0.13] on beam prediction, over 3000 bootstrap draws of the cities, with every one-sided p against a baseline below 0.001.

	LOS/NLOS		Beam prediction (8-antenna input)	
Distance	ρ	gap [metric]	ρ	gap [metric]
Class prior				
Class prior (total variation)	+0.14(4)	33% [0.073]	+0.63(1)	62% [10.5, 44%]
Class prior (KL, directed)	+0.16(4)	33% [0.073]	+0.65(1)	64% [10.3, 44%]
Raw channels				
Label-free SW	+0.27(1)	70% [0.040]	+0.53(1)	62% [10.5, 44%]
Label-aware centroid	+0.19(1)	39% [0.068]	+0.36(1)	19% [14.1, 37%]
Label-aware SW	+0.23(1)	67% [0.043]	+0.33(2)	12% [14.7, 35%]
Joint KL (directed), raw coordinates	+0.33(1)	54% [0.055]	+0.68(1)	67% [10.2, 44%]
DiMTE, one component	+0.20(2)	39% [0.068]	+0.55(2)	52% [11.4, 41%]
OTDD	+0.34(1)	57% [0.052]	+0.67(1)	72% [9.7, 44%]
UMAP, 2-d	+0.12(1)	34% [0.072]	+0.27(1)	−21% [17.4, 27%]
UMAP, 16-d	+0.31(1)	60% [0.049]	+0.32(1)	−21% [17.4, 29%]
UMAP, 2-d, label-supervised	+0.08(1)	20% [0.085]	+0.23(2)	−45% [19.4, 26%]
Task2Vec	+0.22(1)	26% [0.079]	+0.54(2)	51% [11.4, 41%]
LWM embeddings, no source model				
Label-aware centroid	+0.43(2)	65% [0.045]	+0.40(2)	42% [12.2, 39%]
Label-aware SW	+0.46(3)	55% [0.054]	+0.38(2)	37% [12.6, 37%]
Joint KL (directed), LWM coordinates	+0.40(3)	75% [0.036]	+0.62(2)	74% [9.5, 47%]
MMD, label-free	+0.11(4)	50% [0.058]	+0.16(2)	29% [13.3, 35%]
MMD, label-aware	+0.41(3)	65% [0.045]	+0.38(2)	29% [13.3, 36%]
CKA on class centroids	–	–	+0.10(2)	24% [13.7, 33%]
Task2Vec	−0.05(2)	−8% [0.109]	+0.52(2)	31% [13.1, 38%]
Transferability estimators				
LEEP, closed-form source	+0.54(3)	58% [0.051]	+0.29(3)	23% [13.7, 38%]
NCE, closed-form source	+0.48(2)	32% [0.073]	+0.40(2)	29% [13.3, 38%]
OTCE, conditional entropy	+0.21(3)	55% [0.053]	+0.42(2)	26% [13.5, 37%]
OTCE, transport + entropy	+0.17(3)	62% [0.048]	+0.41(2)	36% [12.7, 37%]
LEEP, closed-form source, standardized	+0.71(2)	78% [0.033]	+0.48(2)	31% [13.1, 39%]
NCE, closed-form source, standardized	+0.69(2)	60% [0.049]	+0.52(2)	33% [12.9, 40%]
LEEP on our mixture classifier	+0.73(2)	86% [0.026]	+0.52(3)	35% [12.8, 41%]
NCE on our mixture classifier	+0.73(2)	<u>87%</u> [0.026]	+0.57(2)	40% [12.4, 41%]
LEEP, closed-form source, our projection	+0.72(2)	81% [0.031]	+0.48(2)	30% [13.2, 40%]
NCE, closed-form source, our projection	+0.69(2)	70% [0.041]	+0.52(2)	33% [12.9, 40%]
LEEP on our mixture, our projection	+0.75(2)	<u>87%</u> [0.026]	+0.51(3)	33% [13.0, 40%]
NCE on our mixture, our projection	+0.75(2)	85% [0.027]	+0.56(2)	40% [12.4, 40%]
OTCE, conditional entropy, standardized	+0.29(3)	70% [0.041]	+0.44(2)	20% [14.0, 37%]
OTCE, transport + entropy, standardized	+0.26(3)	77% [0.035]	+0.54(2)	44% [12.0, 39%]
JC-NCE, joint transport	+0.19(3)	36% [0.070]	+0.43(2)	35% [12.7, 37%]
WDJE, bound on our mixture	+0.17(4)	52% [0.056]	+0.07(2)	11% [14.8, 34%]
LEEP, model trained per source	+0.25(1)	74% [0.037]	+0.37(3)	21% [13.9, 39%]
NCE, model trained per source	+0.50(2)	53% [0.055]	+0.45(2)	28% [13.3, 39%]
N-LEEP, model trained per source	+0.33(2)	45% [0.062]	+0.32(2)	21% [13.9, 36%]
LogME, model trained per source	+0.19(2)	50% [0.058]	+0.21(3)	−8% [16.3, 31%]
H-score (regularized), model per source	+0.16(3)	43% [0.064]	+0.20(3)	12% [14.6, 35%]
GBC, model trained per source	+0.40(2)	69% [0.041]	+0.26(2)	15% [14.4, 34%]
TransRate, model trained per source	−0.26(3)	−197% [0.275]	+0.36(2)	6% [15.2, 37%]
PARC, model trained per source	+0.22(4)	40% [0.066]	+0.21(3)	16% [14.4, 34%]

Distance	LOS/NLOS ρ	LOS/NLOS gap [metric]	Beam prediction (8-antenna input) ρ	Beam prediction (8-antenna input) gap [metric]
Learned without transfer supervision				
Self-supervised Deep Sets, LWM embeddings	+0.38(3)	66% [0.044]	+0.37(2)	53% [11.3, 39%]
Learned from the library transfer matrix				
Pairwise transfer predictor, raw descriptors	−0.14(2)	−132% [0.218]	0.00(3)	1% [15.6, 26%]
Pairwise transfer predictor, LWM descriptors	+0.34(2)	67% [0.043]	+0.39(1)	35% [12.8, 36%]
Directed Deep Sets, raw channels	+0.11(1)	37% [0.069]	+0.27(2)	16% [14.3, 34%]
Directed Deep Sets, LWM embeddings	+0.58(3)	61% [0.049]	+0.44(1)	38% [12.5, 37%]
Directed Set Transformer, LWM embeddings	+0.52(3)	45% [0.063]	+0.19(1)	−24% [17.7, 30%]
Baselines that read the target's L1-RSRP sweep				
Directed joint KL, RSRP (1 dB)	–	–	+0.52(2)	68% [10.0, 46%]
Directed joint KL, RSRP (3 dB)	–	–	+0.52(1)	70% [9.9, 47%]
Directed joint KL, RSRP (10 dB)	–	–	+0.50(1)	70% [9.8, 44%]
Class-prior KL, RSRP	–	–	+0.67(1)	65% [10.3, 43%]
Label-aware centroid, RSRP	–	–	+0.50(2)	52% [11.3, 41%]
DiMTE with each source classifier (ours)				
Mixture classifier (default)	**+0.78**(1)	**88%** [0.025]	+0.79(1)	74% [9.5, 43%]
k-NN classifier	+0.76(1)	**88%** [0.025]	+0.79(1)	**81%** [8.9, 47%]
Parzen (kernel) classifier	+0.76(1)	**88%** [0.025]	**+0.82**(1)	80% [9.0, 45%]
One mean per class	+0.65(1)	78% [0.034]	+0.65(1)	49% [11.6, 41%]
DiMTE (learned space)	+0.73(1)	81% [0.031]	+0.70(1)	68% [10.1, 45%]
DiMTE (learned space), validated settings	+0.74(1)	69% [0.041]	+0.79(1)	74% [9.5]
Random source	–	0% [0.102]	–	0% [15.7, 29%]
Genie-aided source	–	100% [0.014]	–	100% [7.4, 50%]
One ordering for every target	+0.67	–	+0.65	–

Table 24: Rank correlation against each deployment model of the measured transfer alone, on both test sets. The raw-CSI CNN shares no features with DiMTE, and the fine-tuned LWM shares its frozen backbone. Per-target Spearman ρ over cross-city pairs. Against the CNN the best baseline of each column is NCE on our mixture classifier and Joint KL (directed), LWM coordinates for LOS/NLOS identification and beam prediction on the DeepMIMO test cities, and LEEP on our mixture classifier and Joint KL (directed), raw coordinates on the Sionna test cities. Best in bold within each block. Shaded rows use the paper's setting.

Distance	DeepMIMO test cities LOS/NLOS	DeepMIMO test cities Beam	Sionna test cities LOS/NLOS	Sionna test cities Beam
Raw-CSI CNN alone, no shared features				
DiMTE	**0.85**	**0.91**	**0.78**	**0.78**
DiMTE (learned space)	0.76	0.76	0.69	0.69
Best of the baselines	0.71	0.77	0.74	0.67
Fine-tuned LWM alone, the backbone of DiMTE				
DiMTE	**0.79**	**0.86**	0.64	**0.76**
DiMTE (learned space)	0.79	0.73	**0.67**	0.68
Best of the baselines	0.69	0.73	0.66	0.65

Table 25: Directedness of DiMTE on both test sets, and on the test locations for camera traps. Per-target Spearman ρ of the estimate $\widehat{P}_{s\to t}$ and of its symmetrized version $\frac{1}{2}(\widehat{P}_{s\to t} + \widehat{P}_{t\to s})$ against the measured transfer, and the paired difference with its 95% bootstrap interval over cities or locations.

Task	directed ρ	symmetrized ρ	difference
DeepMIMO test cities			
LOS/NLOS	0.85	0.71	+0.14 [+0.11, +0.17]
Beam prediction	0.91	0.83	+0.08 [+0.07, +0.09]
UL-AoA sector classification	0.94	0.85	+0.09 [+0.08, +0.10]
Sionna test cities			
LOS/NLOS	0.78	0.55	+0.23 [+0.19, +0.28]
Beam prediction	0.79	0.68	+0.11 [+0.09, +0.12]
Camera-trap test locations			
Camera traps (iWildCam)	0.87	0.83	+0.05 [+0.01, +0.08]

Table 26: Model-based transferability estimators computed on the deployment models of the measured transfer, one fine-tuned LWM and one raw-CSI CNN trained per source (seed 0), on the DeepMIMO test cities. Each estimator reads the source model's outputs on the target's labeled training samples. Per-target Spearman ρ against the measured transfer and share of the random-to-genie gap recovered, with DiMTE for comparison. p is the one-sided city-clustered permutation p-value for DiMTE ranking sources better. Best in bold, second best underlined. Shaded rows use the paper's setting.

	LOS/NLOS			Beam prediction		
Estimator	ρ	gap	p	ρ	gap	p
Fine-tuned LWM per source						
DISCO	+0.29(2)	44%	< 0.001	+0.76(2)	65%	< 0.001
ETran	+0.27(2)	44%	< 0.001	+0.67(2)	44%	< 0.001
GBC	+0.36(3)	28%	< 0.001	+0.54(2)	55%	< 0.001
H-score, regularized	+0.33(3)	47%	< 0.001	+0.71(3)	61%	< 0.001
LEEP	+0.70(4)	<u>92%</u>	< 0.001	<u>+0.78</u>(2)	72%	< 0.001
LogME	+0.31(1)	59%	< 0.001	+0.70(2)	57%	< 0.001
N-LEEP	+0.49(2)	57%	< 0.001	+0.77(2)	74%	< 0.001
NCE	<u>+0.76</u>(3)	82%	< 0.001	<u>+0.78</u>(2)	67%	< 0.001
NCTI	+0.70(2)	88%	< 0.001	+0.65(2)	58%	< 0.001
PARC	+0.59(3)	71%	< 0.001	+0.74(2)	66%	< 0.001
SFDA	−0.05(3)	−43%	< 0.001	−0.53(2)	−79%	< 0.001
TransRate	−0.07(2)	−2%	< 0.001	+0.46(2)	37%	< 0.001
Raw-CSI CNN per source						
DISCO	+0.05(2)	−73%	< 0.001	+0.71(3)	58%	< 0.001
ETran	+0.19(3)	−4%	< 0.001	+0.69(2)	53%	< 0.001
GBC	+0.11(3)	−37%	< 0.001	+0.22(1)	43%	< 0.001
H-score, regularized	+0.41(3)	10%	< 0.001	+0.59(3)	61%	< 0.001
LEEP	+0.75(3)	**93%**	< 0.001	+0.77(2)	<u>75%</u>	< 0.001
LogME	+0.46(2)	46%	< 0.001	+0.56(3)	38%	< 0.001
N-LEEP	+0.45(3)	42%	< 0.001	+0.75(2)	67%	< 0.001
NCE	<u>+0.76</u>(3)	89%	< 0.001	<u>+0.78</u>(2)	72%	< 0.001
NCTI	+0.71(2)	77%	< 0.001	+0.55(2)	52%	< 0.001
PARC	+0.46(2)	29%	< 0.001	+0.70(2)	69%	< 0.001
SFDA	+0.08(2)	−34%	< 0.001	−0.01(3)	13%	< 0.001
TransRate	+0.16(2)	8%	< 0.001	+0.09(1)	35%	< 0.001
DiMTE	**+0.85**(1)	85%	–	**+0.91**(1)	**79%**	–

Table 27: Model-based transferability estimators computed on the deployment models of the measured transfer, one fine-tuned LWM and one raw-CSI CNN trained per source (seed 0), on the Sionna test cities. Each estimator reads the source model's outputs on the target's labeled training samples. Per-target Spearman ρ against the measured transfer and share of the random-to-genie gap recovered, with DiMTE for comparison. p is the one-sided city-clustered permutation p-value for DiMTE ranking sources better. Best in bold, second best underlined. Shaded rows use the paper's setting.

	LOS/NLOS			Beam prediction		
Estimator	ρ	gap	p	ρ	gap	p
Fine-tuned LWM per source						
DISCO	+0.39(1)	72%	< 0.001	+0.62(2)	68%	< 0.001
ETran	+0.26(2)	53%	< 0.001	+0.50(2)	67%	< 0.001
GBC	+0.61(2)	75%	< 0.001	+0.16(2)	14%	< 0.001
H-score, regularized	+0.36(2)	48%	< 0.001	+0.52(3)	45%	< 0.001
LEEP	+0.65(2)	82%	< 0.001	+0.67(2)	**77%**	< 0.001
LogME	+0.37(2)	75%	< 0.001	+0.37(3)	22%	< 0.001
N-LEEP	+0.38(2)	49%	< 0.001	+0.58(2)	61%	< 0.001
NCE	+0.77(1)	77%	0.100	+0.66(2)	69%	< 0.001
NCTI	+0.73(1)	83%	< 0.001	+0.45(2)	37%	< 0.001
PARC	+0.50(3)	65%	< 0.001	+0.37(2)	64%	< 0.001
SFDA	−0.12(2)	−36%	< 0.001	+0.04(3)	4%	< 0.001
TransRate	−0.21(3)	−60%	< 0.001	+0.46(2)	54%	< 0.001
Raw-CSI CNN per source						
DISCO	+0.27(2)	27%	< 0.001	+0.45(2)	47%	< 0.001
ETran	+0.37(3)	15%	< 0.001	+0.50(1)	−9%	< 0.001
GBC	+0.63(2)	70%	< 0.001	+0.20(2)	26%	< 0.001
H-score, regularized	+0.46(2)	−19%	< 0.001	+0.38(3)	39%	< 0.001
LEEP	<u>+0.78</u>(2)	**89%**	0.335	<u>+0.70</u>(2)	73%	< 0.001
LogME	+0.46(2)	79%	< 0.001	+0.16(4)	6%	< 0.001
N-LEEP	+0.42(3)	56%	< 0.001	+0.54(1)	64%	< 0.001
NCE	**+0.85**(2)	84%	1.000	+0.68(2)	70%	< 0.001
NCTI	+0.72(1)	<u>88%</u>	< 0.001	+0.46(2)	32%	< 0.001
PARC	+0.67(2)	82%	< 0.001	+0.11(3)	8%	< 0.001
SFDA	−0.21(3)	−70%	< 0.001	+0.03(3)	17%	< 0.001
TransRate	+0.03(2)	−270%	< 0.001	+0.23(1)	48%	< 0.001
DiMTE	<u>+0.78</u>(1)	<u>88%</u>	–	**+0.79**(1)	<u>74%</u>	–

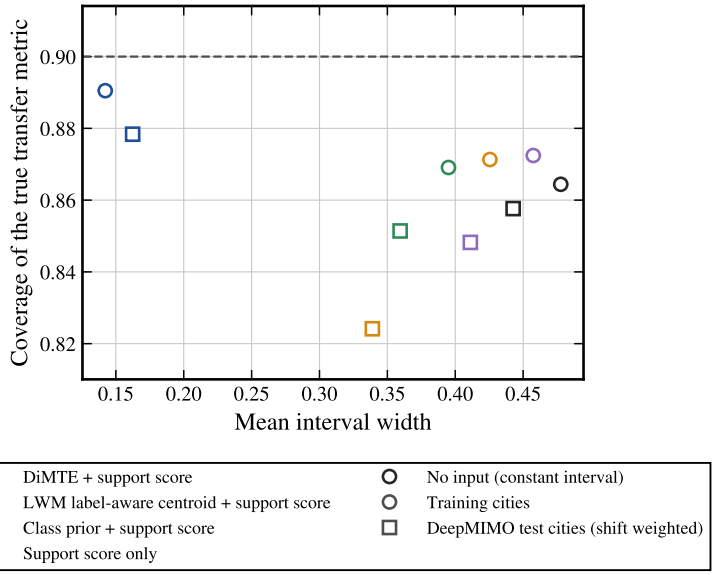
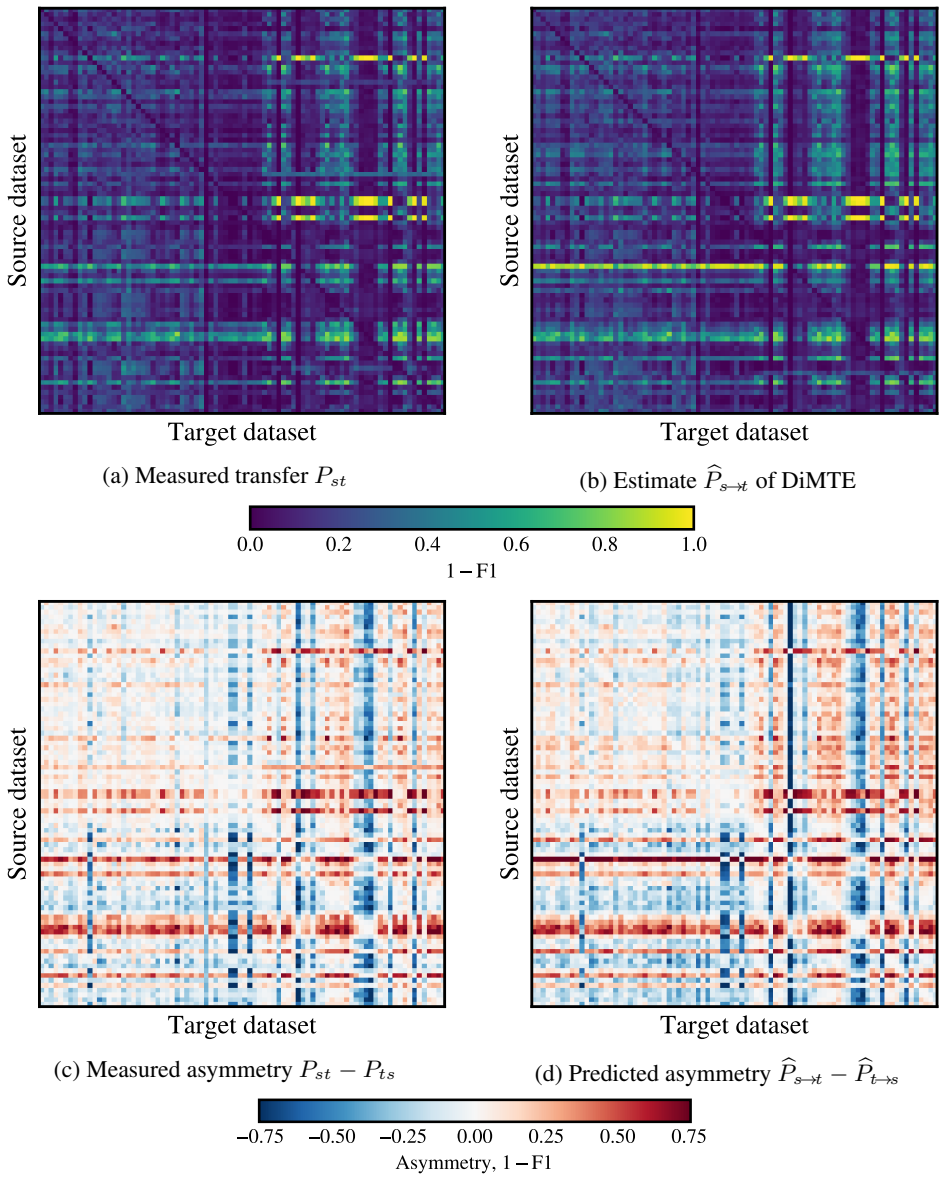


Figure 12: Coverage against mean width of the 90% intervals for LOS/NLOS identification, on the training cities and on the DeepMIMO test cities, for each interval input of Table 36. The estimate of DiMTE comes closest to the dashed nominal rate, with the narrowest intervals of any input. On the Sionna test cities the constant interval covers better (Appendix K.1). Figure 8 shows the point predictions behind the intervals.

Table 28: Each method scored against each of the two deployment models of the measured transfer, per-target Spearman ρ. The estimators score one trained model per source, the raw-CSI CNN or the fine-tuned LWM of the measured transfer, so each agrees most with the deployment model that it reads. DiMTE trains no model per source. The last column scores every method against the deployment model it does not read, the CNN alone for DiMTE and for the estimator on the fine-tuned LWM, and the fine-tuned LWM alone for the estimator on the CNN. Each estimator is the best of its model family against the mean of both deployment models (Tables 26 and 27). Best in bold and second best underlined within each block. Shaded rows use the paper's setting.

	Measured transfer			
Method	Mean of both	CNN alone	Fine-tuned LWM alone	Model not read
DeepMIMO test cities, LOS/NLOS				
DiMTE	**+0.85**	**+0.85**	+0.79	**+0.85**
NCE, CNN per source	+0.76	+0.79	+0.69	+0.69
NCE, fine-tuned LWM per source	+0.76	+0.68	**+0.82**	+0.68
DeepMIMO test cities, Beam prediction				
DiMTE	**+0.91**	**+0.91**	**+0.86**	**+0.91**
NCE, CNN per source	+0.78	+0.78	+0.75	+0.75
NCE, fine-tuned LWM per source	+0.78	+0.74	+0.79	+0.74
Sionna test cities, LOS/NLOS				
DiMTE	+0.78	+0.78	+0.64	**+0.78**
NCE, CNN per source	**+0.85**	**+0.89**	+0.62	+0.62
NCE, fine-tuned LWM per source	+0.77	+0.68	**+0.85**	+0.68
Sionna test cities, Beam prediction				
DiMTE	**+0.79**	**+0.78**	**+0.76**	**+0.78**
LEEP, CNN per source	+0.70	+0.70	+0.66	+0.66
LEEP, fine-tuned LWM per source	+0.67	+0.63	+0.67	+0.63



(a) Measured transfer P_{st}

(b) Estimate $\widehat{P}_{s\to t}$ of DiMTE

(c) Measured asymmetry $P_{st} - P_{ts}$

(d) Predicted asymmetry $\widehat{P}_{s\to t} - \widehat{P}_{t\to s}$

Figure 13: The directed transfer matrix of the 84 LOS/NLOS datasets of the DeepMIMO test cities and the estimate of DiMTE, in the same units, with sources as rows, targets as columns and datasets grouped by city. The antisymmetric parts in (c) and (d) have a Pearson correlation of 0.93 and a Spearman correlation of 0.91 over cross-city pairs. A symmetric score has an antisymmetric part of zero everywhere.

Every comparison is paired over the targets and clustered by city. Two methods are scored on the same targets, and the statistic is the difference of their mean per-target rank correlation, of the share of the gap they recover, or of their regret. Under the null hypothesis that the two methods rank sources equally well, the two method labels are exchangeable. The test therefore exchanges the labels for all base stations of one city at once. The one-sided p-value is the share of $10\,000$ such sign flips that reaches the measured difference, counting the measured assignment itself. A city therefore contributes one unit of evidence however many base stations it holds. The interval resamples the same city clusters 3000 times and reports the percentile 95% interval of the difference.

Table 29: Which classifier the source's labeled embeddings induce, inside the same directed estimate (the expected deployment metric on the target's labeled samples). The number of components, the bandwidth and the number of neighbors are chosen on training cities or locations. Per-target Spearman ρ against the measured transfer, on training-city folds (Train), the DeepMIMO test cities (DeepMIMO), the Sionna test cities (Sionna) and, for camera traps, on training-location folds and the test locations (Test). The last row gives the best of the literature baselines. The mixture contains the one-component summary ($M = 1$) and, with one component per sample and a vanishing variance, the 1-NN rule. Shaded rows use the paper's setting.

	LOS/NLOS			Beam			UL-AoA		Camera traps	
Source classifier	Train	DeepMIMO	Sionna	Train	DeepMIMO	Sionna	Train	DeepMIMO	Train	Test
Mixture (M per class)	0.74	0.85	0.78	0.79	0.91	0.79	0.88	0.94	0.96	0.87
One component per class	0.64	0.74	0.70	0.67	0.77	0.71	0.77	0.87	0.96	0.87
Nearest class mean	0.53	0.60	0.65	0.66	0.79	0.65	0.77	0.87	0.96	0.86
Parzen (Gaussian kernel)	0.76	0.87	0.76	0.80	0.90	0.82	0.88	0.94	0.96	0.88
K-nearest-neighbor rule	0.76	0.87	0.76	0.85	0.91	0.79	0.87	0.93	0.96	0.87
Best literature baseline	–	0.72	0.75	–	0.77	0.68	–	0.79	–	0.81

Table 30: DiMTE on raw channels instead of LWM embeddings. *Normalized* uses the per-sample normalized channel that LWM also reads, and *with received power* uses the scenario channel with its power kept. Each raw feature is evaluated at the paper's setting (standardized principal coordinates with the same k, τ and M as on LWM embeddings), so the representation is the only change, and, where it differs, at the setting chosen on training cities for that feature. Entries are per-target Spearman ρ. The last two rows of each raw block give the paired margin of the LWM mixture over the raw mixture at the paper's setting and its 95% interval. Against the best raw rule of a column the LWM mixture is ahead on LOS/NLOS identification and level or behind on beam. The rows *own settings* use the values chosen on the training cities for that input, for normalized channels, LOS/NLOS $k = 32$, $M = 1$ and beam $k = 32$, $M = 128$. Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction	
Estimate	DeepMIMO test	Sionna test	DeepMIMO test	Sionna test
LWM embeddings				
Mixture classifier (DiMTE)	+0.85	+0.78	+0.91	+0.79
k-NN classifier	+0.87	+0.76	+0.91	+0.79
Parzen classifier	+0.87	+0.76	+0.90	+0.82
Raw channel, normalized				
Mixture classifier, paper's setting	+0.40	+0.21	+0.74	+0.69
Mixture classifier, own settings	+0.29	+0.20	+0.75	+0.69
k-NN classifier	+0.40	+0.21	+0.73	+0.70
Parzen classifier	+0.40	+0.21	+0.73	+0.70
LWM mixture minus this mixture	+0.45	+0.58	+0.16	+0.10
its 95% interval	[+0.41, +0.49]	[+0.53, +0.62]	[+0.12, +0.21]	[+0.08, +0.12]
Raw channel, with received power				
Mixture classifier, paper's setting	+0.26	+0.55	+0.80	+0.76
k-NN classifier	+0.75	+0.72	+0.91	+0.80
Parzen classifier	+0.73	+0.72	+0.89	+0.82
LWM mixture minus this mixture	+0.59	+0.23	+0.10	+0.02
its 95% interval	[+0.55, +0.62]	[+0.20, +0.27]	[+0.09, +0.11]	[+0.02, +0.03]

Inside DiMTE, the two kernel classifiers, the K-nearest-neighbor and Parzen rules, rank sources about as well as the mixture, while one mean per class falls behind. Table 29 keeps the estimate of (3) and changes only the classifier that the source's labeled embeddings induce, in the standardized coordinates of DiMTE. The K-nearest-neighbor rule averages the labels of the K nearest source samples. The Parzen rule sums an isotropic Gaussian kernel over the samples of each class and weighs the classes by their prior, with a bandwidth in multiples of the median nearest-neighbor

distance. Nearest class means use a softmax over negative squared distances. Table 29 covers four sets, namely the DeepMIMO test cities for LOS/NLOS identification, beam prediction and UL-AoA sector classification, and the camera-trap test locations. On each set the training data chose $K = 1$ and the smallest bandwidth of the grid, 0.25 times the median nearest-neighbor distance. With these settings 1-NN reaches 0.87, 0.91, 0.93 and 0.87 on these four sets, against 0.85, 0.91, 0.94 and 0.87 for the mixture. One mean per class reaches only 0.60 and 0.79 there. On the DeepMIMO test cities 1-NN leads on LOS/NLOS identification by +0.02 (interval +0.015 to +0.029), and on beam prediction the two are level (interval -0.004 to +0.009). On the Sionna test cities 1-NN reaches 0.76 against 0.78 on LOS/NLOS identification and 0.79 against 0.79 on beam prediction. On LOS/NLOS identification the mixture stores 15.7 times fewer vectors than samples, so it trades a small loss for a compact summary. Where every source sample can be stored and shared, the 1-NN rule is the better choice on the DeepMIMO test cities and ties the mixture on the Sionna test cities.

The summary reaches the kernel rules' accuracy from a fixed budget of vectors. Figure 15 sweeps the number of components per class and reports both what a source stores and how accurately it estimates the metric. On LOS/NLOS identification the paper's setting stores 64 vectors per dataset against 1000 for the kernel rules, a factor of 15.7, and estimates as well or better (0.043 and 0.038 against 0.044 and 0.045 for the k-NN rule). The curve is flat from 32 components per class, so the training choice sits at the knee rather than at the top of a trade-off. On beam prediction the summaries are larger and the compression is smaller. The paper's setting stores 940 vectors, and the smallest summary within 5% of its error on both sets stores 571 (32 components per class), a factor of 1.8 against the kernel rules. Those rules at 1000 vectors are no better, at 2.1 and 1.9 dB on the DeepMIMO test cities and 2.8 and 2.3 dB on the Sionna test cities, against 1.7 and 2.3 dB. These differences are of the same size as the reproducibility of the measurement itself, 1.3 and 1.4 dB between two halves of its training seeds. The case for the summary is therefore that it reaches that level from a fixed budget of vectors, not that it is more accurate. Table 31 rescores the beam headline at the smaller summaries. On the Sionna test cities a summary of 571 vectors ranks as well as the paper's one and estimates as well, while on the DeepMIMO test cities it costs rank correlation. That is why the choice on the training cities, made on rank correlation, is the larger summary. We keep the choice on the training cities rather than the smaller summary that looks better on the Sionna test set, because that choice, made now, would be a selection on the test set.

The compute that the summary avoids is the training of one model per source. DiMTE ranks the whole library of 84 sources for one task in 15 s on LOS/NLOS identification and 161 s on beam prediction on a CPU, for every pair at once. A single source model takes 4.5 s and 6.5 s to train for the raw-CSI CNN, and 19 s and 86 s for fine-tuned LWM. A library therefore costs 6 to 27 minutes on LOS/NLOS identification and 9 to 120 minutes on beam prediction before any transferability estimator can be scored at all. The kernel variants are faster still, at 1.3 and 1.4 s, because they fit nothing. What they pay is the storage of Figure 15. So the summary buys bounded storage, not speed, and both source classifiers avoid the per-source training that the estimators need. These per-source times, scaled to the library, give the per-seed cost of a transfer matrix quoted in Appendix E.

The average rank of two source classifiers beats the mixture slightly but needs every sample stored (Table 33).

Table 31: Beam prediction with summaries smaller than the paper's setting. Each row keeps DiMTE and its hyperparameters and changes only how many components per class the source summary stores. Rank correlation and gap recovered are scored as in Table 21, and the estimate error is that of Table 39, in dB. The kernel rules store every sample of every source. Shaded rows use the paper's setting.

		DeepMIMO test cities			Sionna test cities		
Source summary	Vectors	ρ	Gap	Est. err.	ρ	Gap	Est. err.
Mixture, $M = 16$	351	+0.87	80%	1.9	+0.79	74%	2.2
Mixture, $M = 32$	571	+0.90	75%	1.7	+0.79	74%	2.3
Mixture, $M = 128$ (paper)	940	+0.91	79%	1.7	+0.79	74%	2.3
k-NN classifier	1000	+0.91	81%	2.1	+0.79	81%	2.8
Parzen (kernel) classifier	1000	+0.90	83%	1.9	+0.82	80%	2.3

Table 32: Which classifier variant the training cities select. The estimate error is the mean absolute difference from measured transfer over cross-city pairs of the training cities, under an isotonic map fitted on the pairs of its other cities. The per-target rank correlation ρ is computed on the same set. Lower error and higher ρ are better, and the best of each column is in bold. The two criteria disagree, which is why the paper reports both and keeps the variant that stores a fixed number of vectors (Figure 15). Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction	
Source classifier	Estimate error	ρ	Estimate error (dB)	ρ
Mixture, M components per class (DiMTE)	**0.031**	+0.74	1.70	+0.79
k-NN classifier	0.032	+0.76	1.93	**+0.85**
Parzen (kernel) classifier	0.032	**+0.76**	**1.68**	+0.80
Nearest class mean	0.061	+0.54	2.71	+0.66

Table 33: DiMTE with each of its source classifiers, and with the ranks of two or three of them averaged per target. The variant is chosen on the training cities, where every setting of this paper is chosen, and applied unchanged to both test sets. Entries are the per-target rank correlation. An average of ranks needs no training and no new summary, only the storage that each classifier already keeps, so the last four rows store every source sample. On the Sionna test cities the training-chosen pair leads the mixture by +0.013 on LOS/NLOS identification and +0.027 on beam prediction. Shaded rows use the paper's setting.

	LOS/NLOS			Beam prediction		
Source classifier	training	DeepMIMO	Sionna	training	DeepMIMO	Sionna
Mixture, M per class (paper)	+0.74	+0.85	+0.78	+0.79	+0.91	+0.79
k-NN classifier	+0.76	+0.87	+0.76	+0.85	+0.91	+0.79
Parzen (kernel) classifier	+0.76	+0.87	+0.76	+0.80	+0.90	+0.82
Mixture and k-NN, ranks averaged	+0.78	+0.88	+0.80	+0.85	+0.91	+0.82
Mixture and Parzen, ranks averaged	+0.78	+0.88	+0.80	+0.80	+0.90	+0.81
k-NN and Parzen, ranks averaged	+0.76	+0.87	+0.76	+0.85	+0.91	+0.82
All three, ranks averaged	+0.78	+0.88	+0.79	+0.84	+0.91	+0.82

Table 34: Where a competing method, or one of our own classifier variants, is ahead of DiMTE. Each row gives the number of DiMTE, the best other number, the method that reaches it, and the interval or the one-sided p where one was computed. Per-target Spearman ρ and the gap recovered (the share of the random-to-genie gap) are better when higher, while $1-$F1 and the L1-RSRP difference in dB are better when lower, and count is the number of the beam baselines a method beats after Holm's correction. Brackets give the 95% interval of the difference. The learned variant is DiMTE (learned space), and m is the number of labeled target samples. One beam row lists a lead of ours that does not survive Holm's correction. The table collects the main adverse results, and the text reports the others next to the results they qualify.

Setting	Metric	Ours	Best other	Method	Interval or p
DeepMIMO test cities					
LOS/NLOS, learned variant	gap	79%	86%	standardized LEEP	-7 points, [-16, -0]
Beam, learned variant	count	45	54	DiMTE	
Beam, learned variant	ρ	0.76	0.91	DiMTE	-0.15, $p = 1.000$
LOS/NLOS, top-3 ensemble	$1-$F1	0.044	0.043	standardized LEEP	$p = 0.675$
LOS/NLOS, planning two models	$1-$F1	0.067	0.066	NCE on our mixture classifier	$p = 0.823$
Sionna test cities					
LOS/NLOS	ρ	0.78	0.85	NCE, CNN per source	$p = 1.000$
Beam	gap	74%	81%	k-NN rule (ours)	
Beam, beam labels only	ρ	0.65	0.68	directed joint KL, raw	-0.03, [-0.06, -0.00]
Beam, pair of classifiers	ρ	+0.816	+0.822	Parzen rule alone	
LOS/NLOS, pair of classifiers	gap	87.7%	88.0%	mixture alone	
Beam, top-3 ensemble	dB	7.70	7.35	directed joint KL on LWM embeddings	$p = 0.900$
Beam, top-5 ensemble	dB	7.31	6.57	directed joint KL on LWM embeddings	$p = 1.000$
Beam, fine-tuned source, m=50	dB	10.0	8.6	best other selector	largest $p = 1.00$
Beam, borrow or train, m=25	dB	9.7	10.2	directed joint KL on LWM embeddings	$p = 0.228$ after Holm

Setting	Metric	Ours	Best other	Method	Interval or p
LOS/NLOS, borrow or train, m=10	1−F1	0.055	0.051	directed joint KL on LWM embeddings	$p = 0.685$
LOS/NLOS, borrow or train, m=25	1−F1	0.031	0.025	NCE on a one-component source	$p = 0.962$
LOS/NLOS, borrow or train, m=50	1−F1	0.023	0.017	LEEP on a one-component source	[0.003, 0.011], Holm $p = 1.000$
Other domains					
Camera traps	ρ	0.87	0.88	Parzen rule (ours)	
MASSIVE locales	gap	66%	75%	MMD, label-aware	
MASSIVE, multilingual-e5 alone	ρ	0.51	0.56	MMD, label-aware	$p = 0.968$

The ranking of DiMTE is specific to the target. One ordering of the sources, read from the other targets' measured transfer, reaches 0.58 and 0.72 on the DeepMIMO test cities and 0.67 and 0.65 on the Sionna test cities, below DiMTE on every set. The ranking is therefore not one library-wide order (the *One ordering for every target* row of Tables 21 and 23). Table 35 removes that source effect from every score and from the measured transfer. DiMTE keeps 0.84, 0.85, 0.76 and 0.80 on the four sets, while the best comparator falls to 0.55, 0.66, 0.59 and 0.66, so the margin grows to between +0.13 and +0.29, every one-sided p <0.001. Most of what the comparators rank is therefore which sources are generally good, while DiMTE ranks what depends on the target.

Table 35: Is the ranking a source main effect? Each score and the measured transfer lose each source's mean over the other targets (other cities only), and the per-target Spearman ρ is recomputed on what remains, the part of transfer that depends on the target. *One ordering* ranks every target by the sources' mean measured transfer to the other targets, the source effect itself. *Best other* is the strongest comparator of each column among all stored methods, before and after centering, and the margin is the paired difference with its 95% interval from the city-clustered tests, where every one-sided p is <0.001. A null control that gives each target the estimate of another city's target keeps a raw ρ of 0.32 to 0.50 and falls to −0.03 to +0.03 after centering.

	Before centering			After centering			
Task	DiMTE	Best other	One ordering	DiMTE	Best other	Margin	95% interval
Sionna test cities							
LOS/NLOS	0.78	0.75	0.67	0.76	0.59	+0.18	[+0.15, +0.21]
Beam	0.79	0.68	0.65	0.80	0.66	+0.13	[+0.10, +0.16]
DeepMIMO test cities							
LOS/NLOS	0.85	0.72	0.58	0.84	0.55	+0.29	[+0.24, +0.33]
Beam	0.91	0.77	0.72	0.85	0.66	+0.19	[+0.16, +0.23]

The comparisons that go against us are reported next to those that favor us. Table 34 collects the main settings where a competing method, or another of our own classifier variants, leads, with the interval or the one-sided p wherever one was computed. Two of them matter for the paper's claims. NCE computed on a trained CNN per source ranks the LOS/NLOS sources of the Sionna test cities better than DiMTE (Table 27). DiMTE (learned space), the learned variant, beats 45 of the 54 beam baselines after a Holm correction (Holm, 1979), while DiMTE beats 54. We report these adverse results next to the results that favor us because a reader cannot weigh one without the other.

A bootstrap that also resamples the candidate sources changes no conclusion. The clusters above carry the dependence among the base stations of one target city, but every target still ranks the same candidate sources. We therefore also resample the 59 DeepMIMO test cities with replacement. Each draw serves both as targets and as candidate sources, so that a target's ρ is recomputed over the drawn sources of other cities. DiMTE then leads the best baseline with an interval of +0.08 to +0.19 on LOS/NLOS identification and +0.09 to +0.18 on beam prediction. DiMTE (learned space) keeps its lead on LOS/NLOS identification (interval +0.01 to +0.16) and stays level on beam prediction.

A number of components chosen on the DeepMIMO test cities themselves would not change the result. We split the DeepMIMO test cities into two random halves 200 times, choose M by the rank correlation on the targets of one half, and score the other half. The chosen M lies between 16 and 64 on LOS/NLOS identification and between 64 and 128 on beam prediction. Against the

best baseline of the first half, DiMTE leads on the second half in 100% and 100% of the splits, with median margins of +0.14 and +0.14 and fifth percentiles of +0.11 and +0.10. It still leads in 100% and 100% of the splits when the baseline is chosen on the scored half itself.

Cached source models, evaluated on the target, rank sources well, but they need a trained model per source and task. If one trained model per source is already stored, a test on the target's labeled samples measures the transfer directly. One seed of the raw-CSI CNN per source, scored against the measured transfer of the remaining seeds, ranks sources at 0.89 on LOS/NLOS identification and 0.92 on beam prediction. One seed of fine-tuned LWM ranks them at 0.88 and 0.90. A cheaper cached model, the MLP on frozen LWM tokens trained once per source, ranks LOS/NLOS sources at 0.82, below DiMTE (0.85). DiMTE is therefore meant for libraries that store data or summaries, and for a new task or deployment model, where no cached models exist.

The L1-RSRP readout helps on both sets, and the Top-1 readout still leads on the DeepMIMO test cities. The Top-1 readout uses the Top-1 error instead of the L1-RSRP difference, so it reads only the beam labels, as the label-only baselines do. With it, DiMTE reaches $\rho = 0.70$ on the training cities and 0.81 on the DeepMIMO test cities, against 0.79 and 0.91 with the L1-RSRP readout. It recovers 76% and 75% of the gap, against 81% and 79%. On the DeepMIMO test cities the Top-1 readout still ranks sources above every baseline (every one-sided p 0.003), while on the training cities the pairwise regressor trained on the transfer matrix leads (Table 19). On the Sionna test cities it reaches 0.65, below directed joint KL on raw channels at 0.68 (difference -0.03, interval -0.06 to -0.00, Table 34).

Most baselines are external, and on LOS/NLOS identification the closest are variants of DiMTE. Of the 49 baselines, 16 score a lightweight source proxy like ours with LEEP, NCE or OTCE, or apply DiMTE to raw channels, and 33 are external methods. On the DeepMIMO test cities the best external baseline reaches 0.60 on LOS/NLOS identification (NCE, model trained per source) and 0.77 on beam prediction (directed joint KL on LWM embeddings). On the Sionna test cities it reaches 0.58 on LOS/NLOS identification (Directed Deep Sets, LWM embeddings). On the DeepMIMO test cities, estimators that read the raw-CSI CNN of every source reach 0.76 and 0.78 (Table 26). On the Sionna test cities, NCE on a CNN trained per source reaches 0.85 on LOS/NLOS identification (Table 27). Among methods without a model per source, the closest comparators on LOS/NLOS identification are LEEP and NCE computed on our source proxy. On beam prediction the strongest is an external score, the directed joint KL on LWM embeddings on the DeepMIMO test cities and the directed joint KL on raw channels on the Sionna test cities (0.68). The comparison isolates what matters in DiMTE, namely the directed estimate in task units of a flexible source classifier (Table 29).

A trained proxy of equal cost ranks sources worse, and the lead of DiMTE does not rest on sites with a rare class. The cheapest trained proxy is a linear classifier trained per source on the same frozen embeddings and evaluated in the task metric on each target. It reaches 0.77 and 0.73, against 0.84 and 0.90 for DiMTE (paired intervals +0.06 to +0.09 and +0.15 to +0.19). Both are scored against seeds 1 and 2 of the measured transfer, because the probe is evaluated on the seed-0 test split. A cached instance of the raw-CSI CNN reaches 0.89 and 0.93 there, which shows what one trained model per source buys. We also keep only the 71 of 84 LOS/NLOS datasets whose minority class holds at least 5% of the samples. DiMTE then still reaches 0.82 against 0.75 for LEEP on our mixture classifier (interval +0.05 to +0.10), so its lead does not come from pairs with a missing class.

The lead is in the ordering of all candidates, not in the single selected source. In the transfer metric of its selected source, DiMTE ties the best baseline on both tasks. Its regret over the genie-aided source is 0.021 in $1-$F1 against 0.020 for LEEP, closed-form source, standardized (paired interval of the difference -0.007 to 0.009, p =0.602), and 1.28 dB against 1.32 dB for directed joint KL on LWM embeddings (p =0.430). When the best of the three top-ranked sources is used, DiMTE recovers 95% and 95% of the random-to-genie gap, against 96% and 88% for the best baselines. Several rules therefore find a good first source. The advantage of DiMTE is that its whole ranking and its predicted metric stay reliable beyond the first entry, which the intervals, few-label selection and region planning rely on.

On the DeepMIMO test cities, model-based estimators evaluated on the deployment models they inspect rank sources below DiMTE. The transferability literature scores a trained model per candidate source. We therefore train the two deployment models of the two-learner mean once per source (seed 0) and compute twelve estimators on each, namely the eight of Appendix E and the more

recent SFDA, NCTI, ETran and DISCO (Table 26). The best is NCE on both deployment models. It reaches 0.76 with the CNN and 0.76 with fine-tuned LWM on LOS/NLOS identification and 0.78 and 0.78 on beam prediction. DiMTE reaches 0.85 and 0.91, with one-sided p below 0.001 in each case. LEEP selects the single best LOS/NLOS source slightly more often and recovers 93% of the gap, against 85%. The two results are consistent, because ρ scores the whole ordering. Estimators designed for many-class image benchmarks, such as SFDA, TransRate and GBC, carry little signal on binary LOS/NLOS identification. All of them need a trained model per source and task, the cost that DiMTE avoids.

On the DeepMIMO test cities DiMTE stays ahead of the closed-form baselines with few target labels. With 10, 20 and 50 labels on beam prediction it reaches 0.70, 0.78 and 0.85, against 0.52, 0.61 and 0.69 for DiMTE (learned space) and 0.62, 0.68 and 0.71 for the best of five closed-form baselines (directed KL divergence between class priors). On LOS/NLOS identification DiMTE reaches 0.63 and 0.79 with 10 and 50 labels, against 0.39 for the best baseline with 10, and DiMTE (learned space) reaches 0.61 and 0.75 (Figure 14a). This first budget study scores LEEP and NCE on a closed-form source, two class-prior distances and the label-aware centroid. Table 41 adds LEEP and NCE on our mixture on the Sionna test cities.

The few-label study trains the raw-CSI CNN on the data that each selection gives. For every target of the DeepMIMO test cities and budget $m \in \{10, 25, 50\}$, m labeled samples are drawn from the target's training split, and each rule selects one source from other cities by reading only those samples. Four arms are scored on the target's test split: (i) a CNN trained on the m samples alone, (ii) the selected source's CNN deployed unchanged, (iii) that CNN fine-tuned on the m samples for 10 epochs at a tenth of the learning rate, with its batch-normalization statistics kept, and (iv) a CNN trained from scratch on the source plus the m samples, oversampled to a fifth of the training set. On LOS/NLOS identification the fine-tuned CNN of the source of DiMTE reaches 0.064, 0.047 and 0.041 with 10, 25 and 50 labels. Target-only training reaches 0.086, 0.050 and 0.035, a random source 0.119, 0.122 and 0.094, and the genie-aided source 0.047, 0.036 and 0.032. Its source also beats the sources that the directed class-prior KL and LEEP on the lightweight source proxy select (0.137 and 0.118 unadapted at 10 labels, against 0.067). On beam prediction, with the L1-RSRP difference in dB, target-only training reaches 18.0, 15.0 and 12.3 with 10, 25 and 50 labels. The source of DiMTE reaches 6.5, 5.7 and 5.6 unchanged, 6.9, 6.4 and 6.2 fine-tuned, and 6.2, 5.5 and 4.7 with joint training. A random source reaches 11.1, 12.3 and 10.5, the class-prior and LEEP sources reach 7.7 and 9.7 at 10 labels, and the genie-aided source reaches 4.3. With a few samples of 32 classes, fine-tuning does not help the selected source, while joint training does. Selection is worth most when the target has few labels. A binary task needs only about 50 labels before its own data are enough, while 32 beams need far more.

On the DeepMIMO test cities, with few labels, DiMTE selects better sources than the strongest competing selectors, and fine-tuning narrows the difference. The few-label study also lets LEEP and NCE on our mixture classifier, standardized LEEP, the directed joint KL on LWM embeddings and the directed class-prior KL select a source from the same m labels. The source of DiMTE, deployed unchanged, beats the best of these on LOS/NLOS identification at 10, 25 and 50 labels. The best competitor reaches 0.085, 0.067 and 0.067, against 0.067, 0.059 and 0.053 for DiMTE, with every $p \leq 0.048$. On beam prediction it beats every competitor with 25 and 50 labels (5.7 and 5.6 dB against 6.5 and 6.4 dB for directed joint KL on LWM embeddings, $p \leq 0.028$) and ties directed joint KL on LWM embeddings with 10 ($p = 0.456$). The differences shrink after fine-tuning, because the fine-tuned model corrects part of a weaker choice. The source of DiMTE then ties LEEP on our mixture classifier on LOS/NLOS identification (p up to 0.389) and directed joint KL on LWM embeddings on beam prediction (p up to 0.721).

An average over the models of several top-ranked sources uses the ranking beyond its first entry. With the CNNs of the one, three and five best-ranked sources averaged per target, DiMTE reaches 5.32, 4.83 and 4.19 dB on beam prediction. The best competing rule reaches 6.22, 5.30 and 4.53 dB ($p = 0.009$, $p = 0.115$ and $p = 0.073$), and the genie-aided ranking 4.31 to 3.33 dB. On LOS/NLOS identification it reaches 0.046, 0.044 and 0.042, level with the best competing rule (0.049, 0.043 and 0.041).

The region-planning experiment places models by facility location. A region is the set of all datasets of randomly drawn DeepMIMO test cities, 23 sites on average. For a budget of R models, a rule picks R prototype sites by greedy facility location on its score matrix. The greedy rule adds

prototypes one at a time, each time the one that most lowers the summed score of every site to its best prototype. A model is trained at each prototype, and every other site uses the prototype with the best score to it. A site's realized metric is the measured transfer from its prototype, averaged over the regions that contain it. The genie-aided plan runs the same greedy rule on the measured matrix. With one model the best baseline (LEEP on our mixture classifier) plans slightly better than DiMTE, with a mean site $1-$F1 of 0.085 against 0.088 and 0.078 for genie-aided planning. With four models the two are level, at 0.053 against 0.054, with 0.040 for genie-aided planning.

K.1 Calibration and Decisions Based on the Estimate

This subsection asks how well the estimate predicts the deployment metric in its own units, and whether that value supports decisions such as reusing a source. The calibration and reuse studies of this subsection were designed after the first Sionna results, so they are exploratory.

Table 36: Calibrated intervals for the transfer metric ($1-$F1 of LOS/NLOS identification, measured transfer) at target coverage 90%, from conformalized quantile regression on the listed inputs. Coverage and mean width over cross-city pairs, with the worst per-city coverage in brackets. Training cities use rotating city folds. The two test sets are fit and calibrated on training cities alone, in the upper block without and in the lower block with the reweighting of the calibration pairs toward the test set's inputs. Shaded rows use the paper's setting.

	Training cities		DeepMIMO test cities		Sionna test cities	
Interval input	coverage	width	coverage	width	coverage	width
Conformalized quantile regression						
DiMTE + support score	0.89 [0.75]	0.142	0.86 [0.48]	0.171	0.82 [0.59]	0.130
Class prior + support score	0.87 [0.45]	0.426	0.79 [0.04]	0.340	0.84 [0.55]	0.318
LWM label-aware centroid + support score	0.87 [0.48]	0.395	0.82 [0.06]	0.364	0.84 [0.57]	0.282
Support score only	0.87 [0.43]	0.457	0.83 [0.06]	0.408	0.85 [0.59]	0.401
No input (constant interval)	0.86 [0.43]	0.478	0.86 [0.19]	0.443	0.85 [0.57]	0.443
The same, reweighted toward the test set's inputs						
DiMTE + support score	–	–	0.88 [0.66]	0.162	0.82 [0.64]	0.129
Class prior + support score	–	–	0.82 [0.18]	0.339	0.84 [0.56]	0.318
LWM label-aware centroid + support score	–	–	0.85 [0.42]	0.359	0.83 [0.56]	0.281
Support score only	–	–	0.85 [0.12]	0.411	0.85 [0.58]	0.399
No input (constant interval)	–	–	0.86 [0.19]	0.443	0.85 [0.57]	0.443

Table 37: Calibrated intervals for beam prediction, for the L1-RSRP difference in dB against the measured transfer, at target coverage 90%. The construction, blocks and columns are those of Table 36, with coverage and mean width over cross-city pairs and the worst per-city coverage in brackets. Shaded rows use the paper's setting.

	Training cities		DeepMIMO test cities		Sionna test cities	
Interval input	coverage	width	coverage	width	coverage	width
Conformalized quantile regression						
DiMTE + support score	0.90 [0.69]	6.9	0.76 [0.35]	5.4	0.74 [0.54]	6.6
Class prior + support score	0.89 [0.39]	11.7	0.74 [0.28]	10.8	0.79 [0.54]	11.1
LWM label-aware centroid + support score	0.90 [0.61]	14.0	0.75 [0.18]	12.9	0.79 [0.50]	13.1
Support score only	0.90 [0.55]	15.7	0.59 [0.06]	13.8	0.75 [0.34]	13.5
No input (constant interval)	0.90 [0.58]	15.5	0.59 [0.29]	14.1	0.78 [0.38]	14.1
The same, reweighted toward the test set's inputs						
DiMTE + support score	–	–	0.85 [0.41]	6.1	0.75 [0.55]	6.4
Class prior + support score	–	–	0.75 [0.28]	11.0	0.76 [0.52]	10.4
LWM label-aware centroid + support score	–	–	0.85 [0.36]	14.6	0.80 [0.50]	13.4
Support score only	–	–	0.62 [0.23]	14.3	0.79 [0.40]	14.6
No input (constant interval)	–	–	0.59 [0.29]	14.1	0.78 [0.38]	14.1

Standardized coordinates also lift LEEP. LEEP on our lightweight source proxy rises from 0.57 to 0.65 on LOS/NLOS identification and from -0.33 to 0.45 on beam prediction when the principal coordinates are standardized. Part of its gain therefore comes from the space. Standardized LEEP ranks LOS/NLOS sources below DiMTE (0.65 against 0.85) but selects a single source about as well. It recovers 86% of the random-to-genie gap against 85%, a difference of -1 points with interval -6 to +5.

On the Sionna test cities a reused source model helps most with few labels, and no rule's single pick is best in every setting. We repeat the few-label study of Appendix K on the Sionna test cities, as an exploratory study designed after the first Sionna results. On beam prediction with 10 labeled target samples, the source that DiMTE selects gives 10.8 dB when deployed unchanged. A CNN trained on those samples alone gives 15.3 dB and a random other-city source 15.4 dB, while joint training on the source and the 10 samples gives 9.7 dB. With 50 samples the source of DiMTE, target-only training and joint training give 9.7, 10.0 and 7.0 dB. The pick of DiMTE is level with the strongest competing selectors at every budget, however, with the largest one-sided p at 0.36, 0.20 and 0.06. In this study LEEP and NCE ran on a one-component source summary, so the 5 selectors are four distinct rules. After fine-tuning, the source that the directed joint KL on LWM embeddings selects is better at every budget (10.0, 8.8 and 8.6 dB against 11.3, 10.6 and 10.0 dB).

On LOS/NLOS identification the picture reverses, because the target's own labels nearly solve the task. At 10 samples, target-only training reaches 0.068 in $1-$F1, against 0.073 for the selected source and 0.061 for the best competing selector. Joint training on the source and the 10 samples reaches 0.047. With 50 samples the pick of DiMTE is ahead of the best competing selector (0.030 against 0.039). An average over the models of several top-ranked sources does not change the verdict either. With the best-ranked source alone, DiMTE reaches 9.52 dB on beam prediction against 9.86 dB for directed joint KL on LWM embeddings ($p = 0.161$). With three and five sources that rule is ahead (7.70 and 7.31 dB against 7.35 and 6.57 dB). On LOS/NLOS identification DiMTE leads with every number of sources without separating (0.024 against 0.025, $p = 0.276$). Every rule stays far ahead of a random other-city source (15.52 dB and 0.095). On the set generated after the method was fixed, several rules therefore solve the choice of one source with few labels equally well, while the ranking of the whole library still separates them.

In task units, the error of DiMTE is smaller than that of a constant predictor on every set. Over cross-city pairs, the estimate misses the measured transfer by 0.046 in $1-$F1 and 3.8 dB on the DeepMIMO test cities, and by 0.049 and 4.4 dB on the Sionna test cities. A constant predictor set to the mean transfer of the training cities misses by 0.117 and 5.8 dB, and 0.088 and 4.8 dB. The error is mostly a bias. The estimate is pessimistic by +3.5 dB on beam pairs of the DeepMIMO test cities and +4.4 dB on the Sionna test cities, because the source-induced classifier is weaker than a trained model. An isotonic map removes that bias. The map is a monotone function from estimate to metric, fitted on the training cities alone, so it trains nothing at the new site. It leaves 1.7 dB and 2.3 dB, against a spread of 6.2 dB and 5.6 dB in the transfer itself. On LOS/NLOS identification the raw estimate is already within 0.046 and 0.049, and recalibration changes little (0.043 and 0.038).

One trained model is enough for most of that calibration on beam prediction. The map above reads every cross-city pair of the training cities. A constant offset can instead be fitted on a single row, namely one seed of the raw-CSI CNN trained on one training dataset and tested on the others. That offset brings the Sionna beam error to 2.5 dB, the median over the 68 choices of that dataset (10th to 90th percentile 2.3 to 3.2 dB), against 4.4 dB raw and 2.3 dB with every pair. On the DeepMIMO test cities it gives 2.0 dB against 3.8 and 1.7 dB. An isotonic map fitted on the same row does worse, at 3.1 dB on Sionna.

With one map fitted on the training cities, DiMTE is ahead of every external score and level with its Parzen variant. A transferability score has no units, so a fair comparison of estimates applies the same map, fitted on the training cities, to every score, including ours. An isotonic map is the most flexible monotone recalibration, so a unitless score gets every chance to match the metric. What remains is the scatter around that curve. A per-target ranking does not control that scatter, because a score can order the sources of each site well and still place different sites on the wrong scale. Table 39 fits that map on the cross-city pairs of the training cities alone, the library the method already assumes, so nothing is fitted on the test set. Of the 28 scores compared there, DiMTE is ahead of every external score on every set and task. It reaches 0.038 in $1-$F1 and 2.3 dB on the Sionna test cities, against 0.054 and 3.1 dB for the best external score, and 0.043 and 1.7 dB on the DeepMIMO test cities, against 0.083 and 3.6 dB. The encoders trained on the transfer matrix of the training cities predict the metric directly and take part without any map. As in the main tables, the standardizing subspace of a score is fitted on the set's unlabeled samples, and only the map from score to metric comes from the training cities.

A calibration fitted on the test set for every method does not overturn the result. Table 38 refits the map for each city on the pairs of the other cities of the same set and admits every method,

Table 38: Every method as an estimate of the deployment metric, under one calibration that none of them fits on the site it is judged on. Each score is mapped to the metric by an isotonic map fitted on the pairs of the other cities of the same set, and the table reports the mean absolute difference from measured transfer over cross-city pairs, in 1−F1 for LOS/NLOS identification and in dB for beam prediction. Lower is better, best in bold and second best underlined among every method scored, not only the rows listed. DiMTE keeps the setting chosen on the training cities, while every other method is scored at the best of its stored variants. Of the 45 to 49 methods scored, the table lists the external method that leads each column and the others with the best mean rank, eight in all, followed by DiMTE with another source classifier. A dash marks a method defined for one task only, such as the baselines that read the target's L1-RSRP sweep. Shaded rows use the paper's setting.

	DeepMIMO test cities		Sionna test cities	
Score, after the leave-one-city-out calibration	LOS/NLOS	Beam (dB)	LOS/NLOS	Beam (dB)
Reproducibility of the measurement	0.042	1.3	0.050	1.4
External methods, each at its best stored variant: the leader of each column and the best mean rank				
Class-prior KL, RSRP-weighted	–	3.2	–	3.1
Directed joint KL, RSRP-weighted (1 dB)	–	2.9	–	3.9
Directed joint KL, RSRP-weighted (3 dB)	–	3.0	–	3.9
Joint KL (directed), LWM coordinates	0.109	2.7	0.070	3.6
Directed Deep Sets, LWM embeddings	0.094	2.9	0.063	4.1
Pairwise transfer predictor, LWM descriptors	0.090	2.4	0.066	4.2
LEEP on our mixture classifier	0.089	3.3	0.042	4.6
LEEP, closed-form source	0.088	4.8	0.043	4.5
The same estimate with another source classifier				
k-NN classifier	**0.039**	**1.6**	0.037	2.5
Parzen (kernel) classifier	0.041	**1.6**	0.037	**2.0**
Nearest class mean	0.074	2.7	0.052	2.9
DiMTE (learned space)	0.048	3.3	0.039	2.7
DiMTE	0.042	**1.6**	**0.033**	2.2

Table 39: Each method as an estimate of the deployment metric, under the calibration an operator can run. An isotonic map fitted on the cross-city pairs of the training cities alone maps each score to the metric, so nothing is fitted on the test cities. Entries are the mean absolute difference from measured transfer over cross-city pairs, in 1−F1 for LOS/NLOS identification and in dB for beam prediction. Lower is better. Best in bold and second best underlined among every method scored, not only the rows listed, and values that print equal share a mark. DiMTE keeps the setting chosen on the training cities. The encoders trained on the transfer matrix of the training cities predict the metric directly and take part without any map. The table lists the external method that leads each column and the others with the best mean rank, eight in all, followed by DiMTE with another source classifier. A dash marks a method defined for one task only, such as the baselines that read the target's L1-RSRP sweep. Methods whose score needs a trained model for every source are compared in Table 38 instead. Shaded rows use the paper's setting.

	DeepMIMO test cities		Sionna test cities	
Score, after the training-fitted calibration	LOS/NLOS	Beam (dB)	LOS/NLOS	Beam (dB)
Reproducibility of the measurement	0.042	1.3	0.050	1.4
External methods: the leader of each column and the best mean rank				
Class-prior KL, RSRP-weighted	–	3.9	–	3.1
Directed joint KL, RSRP-weighted (1 dB)	–	3.6	–	4.1
Directed joint KL, RSRP-weighted (3 dB)	–	3.8	–	4.1
Class prior (KL, directed)	0.115	3.9	0.079	3.4
Class prior (total variation)	0.121	3.9	0.081	3.4
Joint KL (directed), LWM coordinates	0.116	3.6	0.087	4.2
LEEP on our mixture classifier	0.083	4.1	0.068	5.2
LEEP, closed-form source	0.087	5.4	0.054	4.9
The same estimate with another source classifier				
k-NN classifier	0.044	2.1	0.045	2.8
Parzen (kernel) classifier	0.044	1.9	0.045	**2.3**
Nearest class mean	0.077	3.6	0.065	3.1
DiMTE (learned space)	0.048	8.3	0.039	4.1
DiMTE	**0.043**	**1.7**	**0.038**	**2.3**
the same, before the calibration	0.046	3.8	0.049	4.4

including the estimators that need a trained model for each source. DiMTE leads every external method there as well, at 2.2 dB against 3.1 dB on Sionna beam prediction ($p < 0.001$). Among our own classifier variants, the kernel rules are level (2.0 dB). Both numbers sit near the level at which the measurement reproduces itself, 1.4 dB between two halves of its training seeds. A rule that picks the same source can still be wrong about what that source will deliver.

Table 40: Reuse or train, a decision that needs the metric and not an order. At a Sionna test site with m labeled samples, the rule reuses the top-ranked source when its predicted metric is better than what those samples already give, and otherwise trains on them. Every score becomes a predicted metric through an isotonic map fitted on the cross-city pairs of the training cities at the same budget, so a unitless score can take part. All outcomes come from one deployment model, the raw-CSI CNN, and are reported in dB for beam prediction and in $1-$F1 for LOS/NLOS identification, lower is better. The two fixed policies and the best possible decision use the top-ranked source of DiMTE. Shaded rows use the paper's setting.

	Beam prediction (dB)			LOS/NLOS ($1-$F1)		
Policy at a Sionna test site	m=10	25	50	10	25	50
Train on the site's labels	15.3	12.2	10.0	0.068	0.024	0.018
Reuse the top-ranked source	10.8	10.0	9.7	0.073	0.041	0.030
Decide with the best other score	11.1	10.2	9.5	0.047	0.022	0.016
Decide with DiMTE	10.6	9.7	8.8	0.043	0.023	0.017
Best possible decision	10.3	9.2	8.5	0.036	0.018	0.015
Right (%), ours / best other score	89 / 79	81 / 64	77 / 61	74 / 57	70 / 75	76 / 76
Right (%), reusing always	86	74	60	42	25	26
The same decision, with the site's own outcome estimated from its m labels						
Decide with the best other score	11.6	10.5	9.9	0.051	0.025	0.017
Decide with DiMTE	11.1	9.7	9.3	0.055	0.031	0.023
Right (%), ours / best other score	79 / 73	81 / 71	65 / 65	60 / 75	60 / 82	57 / 88

Table 41: Source ranking on the Sionna test cities when each target keeps only m randomly drawn labeled samples. Entries are the per-target Spearman ρ, averaged over 20 draws (one at $m = 1000$, which keeps every sample), with m also given as the mean number of labels per class present in the target (two classes for LOS/NLOS identification, while on beam prediction about 29 of the 32 codewords occur at a site). Sources keep all their samples. A draw in which a rule scores every source equally cannot rank them and counts as $\rho = 0$, a random order. *Gap* is the share of the random-to-genie gap that the source of DiMTE recovers, in the deployment metric. *Best without a source model* is the strongest competing rule that trains no model per source, including LEEP and NCE on our mixture classifier. Its margin comes from a paired test over the targets of all draws, clustered by city, and over all of them the largest one-sided p is 0.120 on LOS/NLOS identification and below 0.001 on beam. *NCE, CNN per source* evaluates the raw-CSI CNN that the measured transfer trains on every candidate source on the drawn samples, which needs one trained model per source.

		DiMTE			Best without a source model		NCE, CNN per source	
m	per class	fixed space	gap	learned space	ρ	margin	ρ	margin
LOS/NLOS								
2	1.0	+0.27	2%	+0.27	+0.23	+0.04 [+0.01, +0.07]	+0.16	+0.11 [+0.09, +0.13]
4	2.0	+0.40	31%	+0.39	+0.36	+0.05 [+0.02, +0.08]	+0.39	+0.01 [−0.01, +0.04]
10	5.0	+0.55	61%	+0.51	+0.54	+0.01 [−0.01, +0.04]	+0.60	−0.05 [−0.06, −0.03]
25	12	+0.66	78%	+0.61	+0.66	+0.01 [−0.00, +0.01]	+0.73	−0.07 [−0.08, −0.06]
50	25	+0.71	82%	+0.66	+0.70	+0.01 [+0.00, +0.02]	+0.79	−0.08 [−0.09, −0.07]
1000	500	+0.78	88%	+0.73	+0.75	+0.03 [+0.01, +0.05]	+0.85	−0.07 [−0.08, −0.05]
Beam prediction								
10	0.4	+0.62	63%	+0.56	+0.57	+0.05 [+0.03, +0.07]	+0.37	+0.25 [+0.21, +0.30]
25	0.9	+0.71	69%	+0.63	+0.62	+0.09 [+0.07, +0.11]	+0.49	+0.22 [+0.18, +0.26]
50	1.8	+0.75	72%	+0.67	+0.63	+0.11 [+0.10, +0.13]	+0.56	+0.18 [+0.14, +0.23]
100	3.6	+0.77	73%	+0.68	+0.64	+0.12 [+0.11, +0.14]	+0.62	+0.15 [+0.11, +0.19]
200	7.2	+0.78	74%	+0.69	+0.65	+0.13 [+0.12, +0.14]	+0.65	+0.13 [+0.09, +0.17]
1000	36	+0.79	74%	+0.70	+0.65	+0.13 [+0.12, +0.15]	+0.68	+0.11 [+0.07, +0.15]

DiMTE leads the rules without a model per source at every budget, and a trained model per source ranks better only on LOS/NLOS identification with five or more labels per class. Table 41 and Figure 15 extend the budget study to one and two labels per class on LOS/NLOS identification

Table 42: What a small site should add to its own data. Each arm trains the raw-CSI CNN on the site's m labeled samples together with the datasets it selects, for the same number of epochs, and is tested on the site's held-out users. The library is the 152 DeepMIMO datasets, so the added data come from another simulator. The comparators are the strongest rules given the same labels in Table 41, LEEP on our mixture classifier for LOS/NLOS identification and the class-prior KL for beam prediction, each adding one and three datasets. Lower is better. Best in bold. Shaded rows use the paper's setting.

		LOS/NLOS (1−F1)		Beam prediction (dB)	
Training data	Added	$m = 100$	$m = 200$	$m = 100$	$m = 200$
The site's samples alone	–	**0.012**$\pm$0.002	**0.007**$\pm$0.002	7.54$\pm$0.60	5.98$\pm$0.46
+ the dataset DiMTE ranks first	1	0.018$\pm$0.003	0.013$\pm$0.002	6.63$\pm$0.43	6.28$\pm$0.52
+ its first three	3	0.021$\pm$0.004	0.016$\pm$0.002	**5.86**$\pm$0.56	**5.66**$\pm$0.50
+ LEEP on our mixture, first	1	0.019$\pm$0.003	0.014$\pm$0.002	7.85$\pm$1.13	6.60$\pm$0.80
+ LEEP on our mixture, first three	3	0.024$\pm$0.005	0.018$\pm$0.003	8.60$\pm$1.44	**5.66**$\pm$0.43
+ class-prior KL, first	1	0.023$\pm$0.003	0.013$\pm$0.003	7.38$\pm$0.84	7.14$\pm$0.76
+ class-prior KL, first three	3	0.033$\pm$0.005	0.018$\pm$0.003	8.19$\pm$1.12	7.42$\pm$1.27
+ three random datasets	3	0.037$\pm$0.004	0.029$\pm$0.006	10.93$\pm$1.01	7.22$\pm$0.74
+ the dataset DiMTE ranks last	1	0.048$\pm$0.005	0.030$\pm$0.004	11.50$\pm$0.88	10.50$\pm$0.81

and to several labels per beam on beam prediction, with twenty draws per budget. A draw in which a rule scores every source equally cannot rank them and counts as $\rho = 0$, a random order. DiMTE scores every source equally in 32% of the LOS/NLOS draws at one label per class, 14% at two and 2% at five, and in none on beam prediction. At every budget DiMTE has the highest mean correlation among the rules that train no model per source. Its margin is +0.05 to +0.13 on beam prediction and +0.01 to +0.05 on LOS/NLOS identification, where LEEP and NCE on our mixture classifier come close. The lead at ten labels is within the noise there (largest p 0.120). At one label per class DiMTE (learned space) is level with it (0.27 against 0.27). NCE on the raw-CSI CNN of every source falls to 0.16 at one label per class, below DiMTE, and is level at two (0.39 against 0.40). It ranks LOS/NLOS sources better from five labels per class on. On beam prediction it needs about 200 labels to reach the ranking that DiMTE reaches with 10. Because NCE scores the CNN itself, we also score it against each deployment model alone. At full labels it reaches 0.89 against the CNN alone and 0.62 against fine-tuned LWM alone (Table 28). In this study LEEP and NCE on our mixture use the projection of the set they score, as DiMTE does. With all labels they reach 0.75 on LOS/NLOS identification, the value of the same variant in Table 23. The margin of DiMTE for LOS/NLOS identification on the Sionna test cities is +0.03 in both studies. The draws are the ones of the earlier three-draw study, which this one extends, so its first three draws reproduce the earlier rows up to ties within rounding.

The choice between reusing a source and training on the site's own labels needs the predicted metric, and DiMTE provides it. At a new site with m labeled samples, an operator can train on those samples or reuse the top-ranked source. The outcome of reuse is unknown before deployment. We decide from the predicted metric alone and reuse the source when its prediction is better than what the m samples already give. Every score takes part through the same isotonic map, fitted on the cross-city pairs of the training cities at the same budget. Every outcome is measured with one deployment model, the raw-CSI CNN (Table 40). What the m samples give is taken as measured, the same for every method, so the decision turns only on the predicted value of the source. A site would have to estimate that side too, so the absolute rates below are optimistic, while the comparison between methods is not. Across the three budgets on beam prediction the decision costs 10.6, 9.7 and 8.8 dB, against 11.1, 10.2 and 9.5 dB for the best competing score under the same map. It is ahead at every budget, and separates with 50 labels ($p = 0.002$, interval $[-1.1, -0.2]$ dB, and $p = 0.010$ after Holm over the 6 budgets and tasks). At 10 and 25 labels the difference is nominal ($p = 0.061$ and $p = 0.046$). On LOS/NLOS identification the ranking of the rules changes with the budget. At 10 labels the decision costs 0.043, the lowest of the rules compared, while NCE on a one-component source decides right more often (81% against 74%). From 25 labels competing scores are ahead on both axes, at 0.022 and 0.016 against 0.023 and 0.017 for DiMTE, and up to 87% and 94% right against 70% and 76%.

The gain over both fixed policies is what the metric buys. The fixed policies of always reusing and always training cost 9.7 dB and 10.0 dB at 50 labels on beam prediction, where the decision reaches 8.8 dB. On LOS/NLOS identification at 10 labels they cost 0.073 and 0.068, where the decision reaches 0.043. The policy of always reusing is also the majority decision. It is right 60% of the

time on beam prediction at 50 labels, against 77% for DiMTE, and 42% against 74% on LOS/NLOS identification at 10 labels. On the accuracy axis the best competing score reaches 76% and 81%. No ordering of the sources can make this decision at all.

On beam prediction the decision holds when the site's own outcome is estimated too. The comparison above takes what the m samples give as measured. A site would have to estimate it. We therefore repeat the decision with that side cross-validated on the m samples alone, with the same CNN recipe, and judge the result against the measured outcomes as before. On beam prediction the decision then costs 11.1, 9.7 and 9.3 dB, against 11.6, 10.5 and 9.9 dB for the best competing score under the same map. It is ahead from 25 labels ($p = 0.016$ and $p = 0.009$), although no cell survives a Holm correction over the 6 cells, where the smallest adjusted value is $p = 0.052$. With 10 labels it is behind the policy of always reusing, which is right 86% of the time there. The decision therefore pays off where the fixed policy is least reliable. The DeepMIMO test set replicates this result. With both sides estimated there too, the decision costs 5.7 and 5.6 dB at 25 and 50 labels against 6.6 and 6.2 dB for the best competing score ($p = 0.001$ and $p = 0.024$). It only matches the policy of always reusing, which is right 94% of the time on that set. On LOS/NLOS identification the estimated side dominates and the rule loses. At 25 and 50 labels it costs 0.031 and 0.023, against 0.024 and 0.018 for training on the site's own labels and 0.025 and 0.017 for the best competing score ($p = 1.000$ at 50 labels, interval [0.003, 0.011]). The competing scores decide right up to 82% and 88% of the time, against 60% and 57% for DiMTE. The same picture holds with the measured side.

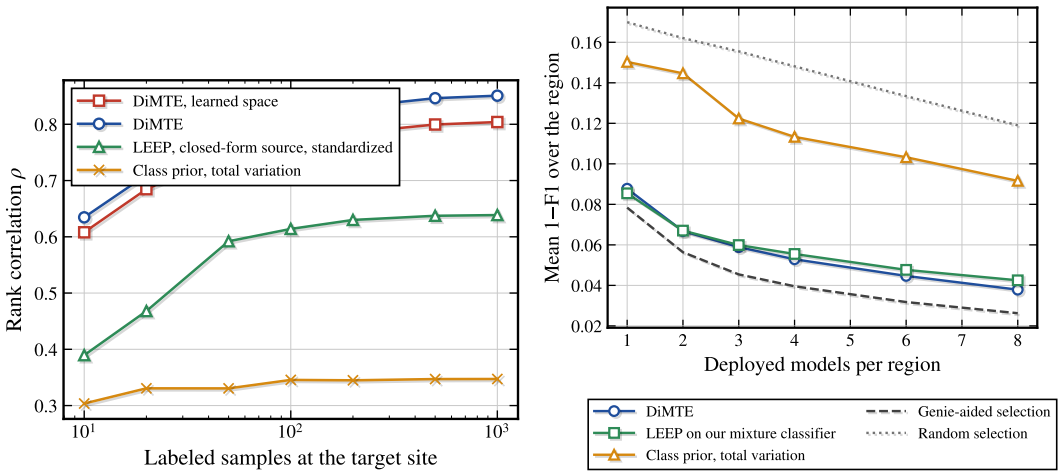
Coverage is marginal over pairs, not conditional on a city. In Tables 36 and 37 the worst city, in brackets, falls well below the nominal rate for every input, the constant interval included. Coverage also falls short of 90% over pairs on both test sets. On the Sionna test cities the constant interval comes closer, at 0.85 against 0.82 on LOS/NLOS identification and 0.78 against 0.74 on beam prediction, while it is 0.443 and 14.1 dB wide, against 0.130 and 6.6 dB for DiMTE. On that set the calibration buys width, not coverage. Pairs of one city share its geometry, so their errors move together. Coverage per city would need calibration pairs from that city, which a new site does not have before any candidate deployment model is trained there. The interval therefore describes the typical pair of a library, and a site with a high support score, the share of its samples outside the source support (Appendix E), calls for caution.

Calibration carries over between the two deployment models. We calibrate on one deployment model and test coverage on the other. Intervals calibrated on the CNN and tested on fine-tuned LWM cover 0.90 on the training cities and 0.88 on the DeepMIMO test cities. With the CNN for both calibration and test, coverage is 0.89 and 0.90. In the reverse direction, coverage is 0.81 and 0.85, compared with 0.90 and 0.89. A switch between the two deployment models costs at most 9 points of coverage and leaves the interval width unchanged.

A user's own transfer measurements do not improve on our calibration. An operator might train one source model and measure it on m of its own sites, which gives one row of a private transfer matrix. We replace our conformal correction with that row and score the other pairs against the user's deployment model. Coverage is then 0.83 at m=10, 0.81 at m=20 and 0.82 at m=40, compared with 0.87 when our training calibration is transferred unchanged. At m=5, coverage falls to 0.70 with standard deviation 0.15 across rows, as conformal prediction expects, because at a 90% target no finite correction is valid with fewer than nine calibration points. On beam prediction the same row gives 0.83 at m=20 against 0.83 unchanged. The practical recommendation is to keep the training calibration.

K.2 Sources Labeled with a Coarser Codebook

The main text assumes the source and the target share a label set. Beam prediction gives the mismatched case a physical form that 3GPP already uses. A site that only ever swept a coarse codebook holds labels over G wide beams, while the target deploys the fine 32-beam codebook. The discrete Fourier transform codebook is ordered by angle, so a group of contiguous beams forms a coarser angular codebook rather than an arbitrary relabeling. The raw-CSI CNN trains on the source's own G labels. At the target, the CNN predicts a wide beam and refines it to one fine beam by the rule a deployment would use. That rule picks the fine beam of the group with the largest mean power in the source's own data, so it reads nothing from the target. The delivered metric is the L1-RSRP difference of that beam on the target's held-out users, measured against the genie-aided beam among



(a) Labels the new site needs

(b) Models a region needs

Figure 14: Two questions an operator asks, answered on the DeepMIMO test cities for LOS/NLOS identification. (a) Rank correlation of each score when only a few labeled samples are available at the target site, as at a new deployment. (b) Mean 1−F1 over a region of 23 sites when R prototype models are deployed and every site uses its best prototype by the score. The genie-aided plan and random prototypes are the two limits. DiMTE and its learned variant still rank sources at $\rho = 0.67$ and 0.64 when a classifier trained on the library predicts the target labels (92% correct), close to the strongest baseline with true labels (0.72).

all 32. Every source is therefore judged on the same fine task, however coarse its labels were. DiMTE evaluates the coarse proxy under the same fine loss, $\widehat{P}_{s\to t} = \mathbb{E}_x \sum_c p_s(c \mid x)\, \ell_t(x, \mathrm{refine}_s(c))$, so only the loss vector changes and the estimator does not.

A coarse label set costs the site little, and DiMTE still ranks the sources that carry it. Over 152 library sources and 20 target sites, a source labeled with 4-beam groups delivers 20.4 dB, where the same source with the full codebook delivers 17.9 dB. The coarse labels thus cost 2.4 dB, and groups of 2 beams bring that cost to 0.9 dB. DiMTE ranks those coarse sources at $\rho = 0.67$ and 0.70 and recovers 71% and 64% of the random-to-genie gap. The source it selects therefore delivers 13.8 dB and 12.8 dB against 20.4 dB and 18.9 dB for a random one. No comparator appears here, because a class prior or a label-aware distance is undefined between two different label sets without a map between them. The readout of DiMTE needs only the loss that the target measures, so it needs no such map.

L Task-Specific Pseudo-Labels

This appendix asks how DiMTE works when the target has no labels or only a few. The pseudo-label studies here were designed after the first results on the test cities, so they are exploratory.

With no target label, DiMTE reads pseudo-labels and still ranks sources ahead of every label-free rule. Every rule that reads target labels can read pseudo-labels instead, while the sources keep their own. For LOS/NLOS identification, an MLP on frozen LWM tokens, trained on the labels of the training cities alone, labels the targets of the DeepMIMO and Sionna test cities with 93% and 97% accuracy. DiMTE ranks the sources of the DeepMIMO and Sionna test cities at 0.66 and 0.68 on these pseudo-labels, ahead of the best label-free rule by +0.38 [+0.32, +0.43] and +0.30 [+0.22, +0.37] (Table 43). With all labels it reaches 0.85 and 0.78. LEEP and NCE on our mixture, given the same pseudo-labels, tie DiMTE on the Sionna test cities and rank below it on the DeepMIMO test cities. For beam prediction, the target's own eight-antenna sub-array, whose channel is the model input, can sweep the codebook directions coarsely with the first eight elements of every codeword. That sweep needs no label, no training and no full-array measurement, and it picks the best codeword for 78% and 83% of target samples in the two test sets. DiMTE ranks sources at 0.89 and 0.75 on these pseudo-labels, against 0.70 and 0.53 for the best label-free rule and 0.91 and 0.79 with all labels. The sweep has no parameter to choose.

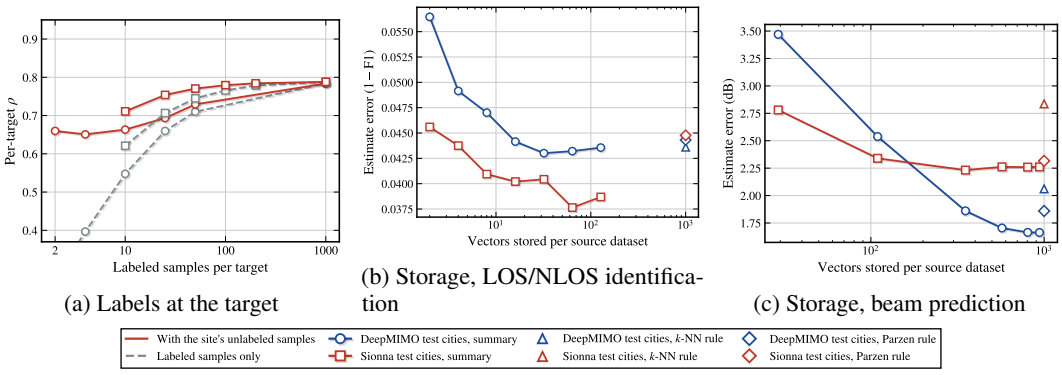
With a few labels, the labeled samples correct the pseudo-labeled ones. Let U_t hold all n samples of a target and $V_t \subset U_t$ a uniform draw of m of them with true labels. Every sample of U_t carries a pseudo-label, and on beam prediction it also carries the losses $\tilde{\ell}_i(c)$ of the sub-array sweep, which replace the measured $\ell(c, y_i, a_i)$. For an additive loss the estimate of Section 6.2 is the prediction-powered mean (Angelopoulos et al., 2023)

$$\widehat{P}^{\mathrm{PP}}_{s\to t} = \frac{1}{n}\sum_{i\in U_t}\sum_{c} p_s(c \mid u_i)\,\tilde{\ell}_i(c) + \frac{1}{m}\sum_{i\in V_t}\sum_{c} p_s(c \mid u_i)\,\big[\ell(c, y_i, a_i) - \tilde{\ell}_i(c)\big]. \tag{10}$$

For an additive loss the second term removes the average error of the pseudo-labels. The estimate therefore stays unbiased whatever their quality, and poor pseudo-labels cost variance rather than bias. On LOS/NLOS identification the pseudo-label takes the place of y_i. Weighted F1 is not additive, so the correction applies entry by entry to the soft confusion matrix of Section 4.1, whose entries are sample means. Each corrected entry is then clipped at zero. The corrected entries are unbiased, but the F1 formed from them is not.

Every beam-prediction study reads one of four kinds of target measurement. (i) The headline results and the label-budget study read, for each labeled user, its best-beam label and its full 32-beam L1-RSRP vector, as one exhaustive sweep of those users provides. (ii) The studies with the site's unlabeled users add the 8-antenna sub-array sweep of every user to the same labeled vectors. (iii) The zero-label ranking of Table 43 reads the sub-array sweep alone, and the full-array powers enter only the reported accuracy of the pseudo-labels. (iv) The Top-1 readout of Appendix K reads the best-beam labels alone.

The corrected estimate is robust to sweep noise and to missing pseudo-labels, while label flips lower it to about the level of the labeled samples alone. We perturb the pseudo-labels of the Sionna test cities at ten labeled samples, with the draws of Table 44. On beam prediction, Gaussian noise of 1, 3 and 6 dB on the power of every swept beam lowers the Top-1/1 accuracy of the pseudo-labels from 83% to 25%. The ranking moves only from 0.71 to 0.68, above the 0.62 of the labeled samples alone. The pseudo-loss vectors keep the order of the beams under noise that changes their best beam. When only half or a quarter of the unlabeled users have a sweep, DiMTE still reaches 0.71 and 0.71. On LOS/NLOS identification, flips of a further 5, 10 and 20% of the pseudo-labels lower their accuracy to 78% and the ranking from 0.66 to 0.55, which matches the 0.55 of the labeled samples alone. Pseudo-label errors then add variance but no bias, as (10) predicts. Missing pseudo-labels cost little (0.65 with one user in four). The pseudo-labels need no measurement beyond what the site already has. On LOS/NLOS identification a classifier trained on the library labels the site's own channels, and on beam prediction the sub-array channel that the deployment model reads is beamformed toward the 32 codebook directions.



(a) Labels at the target

(b) Storage, LOS/NLOS identification

(c) Storage, beam prediction

Figure 15: What DiMTE needs at the target and what it keeps per source. (a) Mean per-target correlation on the Sionna test cities against the number of labeled target samples, over twenty draws, with the site's unlabeled channels (solid) and with the labeled samples alone (dashed). Circles are LOS/NLOS identification and squares beam prediction. (b,c) Mean absolute difference between predicted and measured transfer over cross-city pairs, under the map of Table 39 fitted on the training cities, against the vectors a source stores. The mixture stores $\sum_c \min(M, n_{sc})$ vectors, and the kernel rules store every sample. Lower is better on both axes. The paper's summaries store 64 and 940 vectors, 15.7 and 1.1 times fewer than the samples they replace.

Table 43: Source ranking with no target label. Each target is pseudo-labeled, and the label-aware rules read those labels while sources keep their own. For LOS/NLOS identification the pseudo-labels come from a classifier trained on the labels of the training cities only. For beam prediction the *sub-array sweep* scores the 32 codebook directions with the first eight elements of every codeword on the target's own sub-array channel, the model input, and needs no label, training or full-array measurement. Pseudo-label accuracy is the share of target samples whose class, or best codeword, is correct. Per-target Spearman ρ. The margin of DiMTE over the best label-free method of Tables 20 and 23 comes from a paired test clustered by city. Shaded rows use the paper's setting.

Rule	DeepMIMO test	Sionna test
LOS/NLOS, trained pseudo-labels		
Pseudo-label accuracy	93%	97%
DiMTE	+0.66	+0.68
LEEP, our mixture classifier	+0.56	+0.68
NCE, our mixture classifier	+0.55	+0.68
Class prior (KL, directed)	+0.29	+0.18
Best label-free rule	+0.29	+0.38
DiMTE over it	+0.38 [+0.32, +0.43]	+0.30 [+0.22, +0.37]
DiMTE, all target labels	+0.85	+0.78
Beam prediction, sub-array sweep		
Pseudo-label accuracy	78%	83%
DiMTE	+0.89	+0.75
LEEP, our mixture classifier	+0.73	+0.52
NCE, our mixture classifier	+0.74	+0.56
Class prior (KL, directed)	+0.73	+0.65
Best label-free rule	+0.70	+0.53
DiMTE over it	+0.19 [+0.15, +0.23]	+0.22 [+0.20, +0.24]
DiMTE, all target labels	+0.91	+0.79

Table 44: Source ranking when the target's unlabeled samples are read as well, per-target Spearman ρ over twenty draws, where a draw in which a rule scores every source equally counts as $\rho = 0$. Every unlabeled sample carries a pseudo-label, from the classifier of Appendix L for LOS/NLOS identification and from the coarse sub-array sweep for beam prediction, and the labeled samples correct the estimate (Angelopoulos et al., 2023). The rules that read target labels get the same samples. For the class-prior KL the correction applies to the class frequencies. The recipe was added after the first Sionna results, so this study is exploratory. It was chosen on the training cities, where it is at least as good as the labeled samples alone at every budget of both tasks. Best of each column in bold, second best underlined. Shaded rows use the paper's setting.

LOS/NLOS Labeled samples per target	2	4	10	25	50	1000
Training cities						
DiMTE, labeled samples only	+0.21	+0.31	+0.43	+0.55	+0.61	**+0.74**
DiMTE, with the site's unlabeled samples	**+0.51**	**+0.49**	**+0.52**	**+0.58**	**+0.62**	**+0.74**
Best of LEEP and NCE on our mixture, labeled only	+0.08	+0.22	+0.37	+0.51	+0.57	+0.67
with the site's unlabeled samples	+0.47	+0.43	+0.44	+0.51	+0.56	+0.67
Class-prior KL, labeled only	+0.13	+0.16	+0.18	+0.19	+0.20	+0.20
with the site's unlabeled samples	+0.18	+0.18	+0.19	+0.20	+0.20	+0.20
DeepMIMO test cities						
DiMTE, labeled samples only	+0.43	+0.52	+0.64	+0.73	+0.78	**+0.85**
DiMTE, with the site's unlabeled samples	**+0.65**	**+0.66**	**+0.70**	**+0.76**	**+0.79**	**+0.85**
Best of LEEP and NCE on our mixture, labeled only	+0.11	+0.25	+0.43	+0.55	+0.62	+0.71
with the site's unlabeled samples	+0.46	+0.44	+0.48	+0.53	+0.57	+0.72
Class-prior KL, labeled only	+0.26	+0.28	+0.32	+0.34	+0.35	+0.36
with the site's unlabeled samples	+0.30	+0.32	+0.34	+0.35	+0.35	+0.36
Sionna test cities						
DiMTE, labeled samples only	+0.27	+0.40	+0.55	+0.66	+0.71	**+0.78**
DiMTE, with the site's unlabeled samples	**+0.66**	**+0.65**	**+0.66**	**+0.69**	**+0.73**	**+0.78**
Best of LEEP and NCE on our mixture, labeled only	+0.15	+0.36	+0.54	+0.65	+0.70	+0.75
with the site's unlabeled samples	+0.65	+0.64	+0.65	+0.68	+0.72	+0.75
Class-prior KL, labeled only	+0.13	+0.11	+0.14	+0.16	+0.15	+0.16
with the site's unlabeled samples	+0.16	+0.16	+0.16	+0.16	+0.16	+0.16

Beam prediction Labeled samples per target	10	25	50	100	200	1000
Training cities						
DiMTE, labeled samples only	+0.58	+0.68	+0.73	+0.76	**+0.78**	**+0.79**
DiMTE, with the site's unlabeled samples	**+0.67**	**+0.73**	**+0.76**	**+0.78**	**+0.78**	**+0.79**
Best of LEEP and NCE on our mixture, labeled only	+0.19	+0.29	+0.36	+0.42	+0.47	+0.53
with the site's unlabeled samples	+0.51	+0.51	+0.51	+0.52	+0.52	+0.53
Class-prior KL, labeled only	+0.58	+0.64	+0.67	+0.68	+0.68	+0.69
with the site's unlabeled samples	**+0.67**	+0.68	+0.68	+0.69	+0.69	+0.69
DeepMIMO test cities						
DiMTE, labeled samples only	+0.70	+0.80	+0.85	+0.88	+0.89	**+0.91**
DiMTE, with the site's unlabeled samples	**+0.77**	**+0.84**	**+0.87**	**+0.89**	**+0.90**	**+0.91**
Best of LEEP and NCE on our mixture, labeled only	+0.35	+0.52	+0.61	+0.67	+0.71	+0.74
with the site's unlabeled samples	+0.73	+0.73	+0.74	+0.74	+0.74	+0.74
Class-prior KL, labeled only	+0.62	+0.68	+0.71	+0.72	+0.73	+0.73
with the site's unlabeled samples	+0.72	+0.73	+0.73	+0.73	+0.73	+0.73
Sionna test cities						
DiMTE, labeled samples only	+0.62	+0.71	+0.75	+0.77	**+0.78**	**+0.79**
DiMTE, with the site's unlabeled samples	**+0.71**	**+0.75**	**+0.77**	**+0.78**	**+0.78**	**+0.79**
Best of LEEP and NCE on our mixture, labeled only	+0.35	+0.43	+0.49	+0.53	+0.55	+0.56
with the site's unlabeled samples	+0.55	+0.55	+0.56	+0.56	+0.56	+0.56
Class-prior KL, labeled only	+0.57	+0.62	+0.63	+0.64	+0.65	+0.65
with the site's unlabeled samples	+0.64	+0.65	+0.65	+0.65	+0.65	+0.65

M Scaling with the Number of Datasets and the Summary Size

Two quantities decide what an operator has to collect, namely how many datasets the library holds and how many samples summarize each of them. The first studies vary the number of candidate sources on both tasks, first within the sets of the main text and then on a library of all 236 datasets. The last two vary the training library of DiMTE (learned space) and the summary size, on LOS/NLOS identification in the DeepMIMO test cities. Every study except the one on all 236 datasets scores against the two-learner mean of Section 5.2.

Ranking quality is flat in the number of candidate sources a target ranks. A score is useful only if it keeps ordering a library that grows. We therefore subsample the candidate sources of every target from the sets of the main text and read the per-target rank correlation and the gap recovered against the number of candidates. The correlation is flat or rises. On the DeepMIMO test cities it goes from +0.83 with 20 candidates to +0.84 with 80 on LOS/NLOS identification and from +0.89 to +0.91 on beam prediction. On the Sionna test cities it goes from +0.77 to +0.78 and from +0.76 to +0.79. The gap recovered by the single pick moves the other way on three of the four. It goes from 85% to 83% on DeepMIMO beam prediction. On the Sionna test cities it goes from 86% to 83% on LOS/NLOS identification and from 77% to 74% on beam prediction. LOS/NLOS identification on the DeepMIMO test cities rises from 82% to 86%. So the ordering holds as the candidate set grows, while the single pick gets slightly harder.

On a library of 236 datasets, DiMTE leads every external score. We pool every wireless set of this paper into one library of 236 datasets in 145 cities. One seed of the raw-CSI CNN measures a transfer matrix over all of them. The CNN trains on 800 samples of each source and is tested on 200 samples of each target. The cross-city transfer of that matrix is 0.17 in $1-$F1, the level of the sets in the main text, so the matrix measures the deployment model that DiMTE stands in for. We then rank the same candidates with 12 closed-form scores of the main text, including the strongest comparator of each task on the Sionna test cities. Every dataset serves as a target, and we report the 168 targets in the test cities, where no setting was chosen. DiMTE and every comparator read all 1000 samples of a target, including the 200 that test the CNN. DiMTE rises from 0.73 with 20 candidate sources per target to 0.76 with 230 and recovers 86% of the gap. It leads all 12 external scores at every size, by +0.056 over the strongest, NCE on our mixture at 0.70, with a city-clustered interval of [+0.039, +0.076] and every one-sided p below 0.001 (Table 45). Over all 236 targets, training cities included, it reaches 0.75. Our own classifier variants rank higher on this task. The pair chosen on the training cities reaches 0.79 and the kernel rules 0.78, the same ordering as on the training and DeepMIMO test cities (Table 33).

On beam prediction, DiMTE also leads every external score on that library. A second matrix over the same 236 datasets measures sub-array beam prediction with the same deployment model and split. Its cross-city transfer is 15.6 dB. DiMTE rises from 0.80 with 20 candidate sources to 0.81 with 230 and recovers 68% of the gap. It leads all 12 external scores, by +0.193 [+0.159, +0.229] over the strongest, NCE on our mixture at 0.62, with every one-sided p below 0.001. Its classifier variants rank within about a hundredth of it, at 0.81 for the Parzen rule and 0.80 for the pair, and their single picks are better. The gap recovered by the mixture's pick falls from 77% with 20 candidates to 68% with 230, against 71% and 75% for the two variants (Table 45). The cross-simulator ranking of Section 6.1 reads one block of this matrix. Its sources are the 152 datasets of the training and DeepMIMO test cities, and its targets are the 84 Sionna datasets. There the principal directions are fitted on the library's samples alone, so no target sample enters the projection.

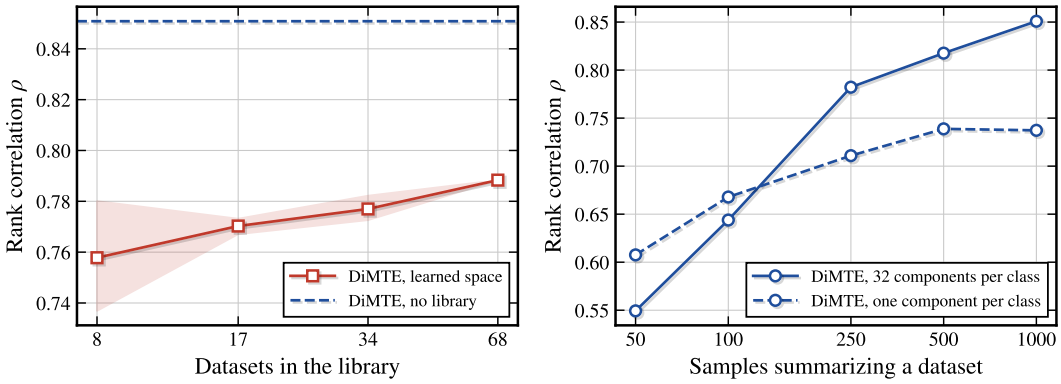
DiMTE (learned space) needs few library datasets, but no library the training cities provide, up to all sixty-eight datasets, lifts it to the level of DiMTE. We train it on a city-disjoint subsample of the training cities and apply it unchanged to the DeepMIMO test cities. Averaged over three subsamples, it reaches $\rho = 0.76$ from eight datasets, 0.77 from seventeen, 0.78 from thirty-four and 0.79 from all sixty-eight, against 0.85 for DiMTE. The numbers here use a three-seed ensemble, which is why the full-library value differs slightly from the five-seed result of Table 20.

The mixture summary needs a few hundred samples per dataset. Every dataset, as source and as target, keeps only its first n_s samples. On LOS/NLOS identification in the DeepMIMO test cities DiMTE reaches $\rho = 0.55$ at n_s=50, 0.64 at 100, 0.78 at 250, 0.82 at 500 and 0.85 at 1000. With one component per class it reaches 0.61, 0.67, 0.71, 0.74 and 0.74. Several components need samples to place their centroids, so one component is better below about a hundred samples per dataset, and

Table 45: A library of 236 datasets in 145 cities, pooled from every wireless set of this paper. One seed of the raw-CSI CNN trains on 800 samples of each source and is tested on 200 samples of each target. Each cell gives, over the 168 targets in the test cities, the per-target rank correlation and the share of the random-to-genie gap that the single pick recovers, at that many candidate sources per target, averaged over three draws at every size. *Pair* ranks sources by the mean rank of the two classifiers chosen on the training cities (Table 33). The matrices transfer at 0.17 in $1-\mathrm{F1}$ and 15.6 dB in L1-RSRP difference across cities, the level of the sets in the main text. Shaded rows use the paper's setting.

Score	20	40	80	160	230
LOS/NLOS identification: DiMTE and its classifier variants					
DiMTE	+0.73/85%	+0.75/86%	+0.75/86%	+0.76/85%	+0.76/86%
1-NN classifier	+0.75/88%	+0.77/88%	+0.77/88%	+0.78/88%	+0.78/89%
Parzen (kernel) classifier	+0.75/88%	+0.77/88%	+0.77/88%	+0.78/88%	+0.78/89%
Pair chosen on the training cities	+0.76/87%	+0.78/88%	+0.78/88%	+0.79/86%	+0.79/87%
The six strongest of the 12 external scores					
NCE on our mixture	+0.67/80%	+0.69/82%	+0.70/82%	+0.70/82%	+0.70/82%
LEEP on our mixture	+0.66/79%	+0.68/79%	+0.69/79%	+0.69/78%	+0.69/81%
Standardized NCE	+0.55/58%	+0.56/63%	+0.57/60%	+0.57/62%	+0.57/60%
Standardized LEEP	+0.55/65%	+0.55/69%	+0.56/62%	+0.56/66%	+0.56/68%
Class-prior KL, directed	+0.26/31%	+0.24/27%	+0.25/29%	+0.25/29%	+0.25/29%
Class-prior total variation	+0.25/31%	+0.24/26%	+0.24/29%	+0.25/29%	+0.25/30%
Beam prediction: DiMTE and its classifier variants					
DiMTE	+0.80/77%	+0.80/77%	+0.81/72%	+0.81/67%	+0.81/68%
1-NN classifier	+0.75/75%	+0.76/77%	+0.77/75%	+0.77/77%	+0.77/79%
Parzen (kernel) classifier	+0.79/78%	+0.79/76%	+0.80/73%	+0.81/71%	+0.81/71%
Pair chosen on the training cities	+0.78/77%	+0.79/78%	+0.80/76%	+0.80/75%	+0.80/75%
The six strongest of the 12 external scores					
NCE on our mixture	+0.62/58%	+0.61/58%	+0.62/57%	+0.62/58%	+0.62/57%
LEEP on our mixture	+0.60/52%	+0.60/50%	+0.60/45%	+0.61/45%	+0.61/44%
Class-prior KL, directed	+0.56/52%	+0.57/54%	+0.58/54%	+0.58/52%	+0.58/52%
Standardized LEEP	+0.55/60%	+0.56/58%	+0.56/60%	+0.57/58%	+0.57/58%
Standardized NCE	+0.56/55%	+0.57/55%	+0.57/54%	+0.57/55%	+0.57/53%
Directed joint KL, LWM embeddings	+0.56/63%	+0.56/62%	+0.57/63%	+0.57/60%	+0.57/61%

the mixture leads from 250 on. The cost of comparing two sites therefore stays bounded by a short measurement campaign.



(a) Library size (b) Summary size

Figure 16: Scaling on the DeepMIMO test cities, LOS/NLOS identification. (a) DiMTE (learned space) trained on a city-disjoint subsample of L training datasets, with the band covering three subsamples. (b) DiMTE when every dataset is summarized by n_s samples.

N A Variant with a Learned Embedding Space

DiMTE fits its projection A by principal components, which read no transfer outcome. When measured transfer between training cities is available, a variant can learn the embedding space from it. This appendix describes that variant and shows that it does not improve on the fixed space on the wireless tasks (on MASSIVE it does, Appendix P). The variant, DiMTE (learned space), learns three parts from the transfer matrix of the training cities: (i) a residual projection $A(z - \bar{z}) + h(z)$, with A initialized at the principal directions and h a two-layer network with zero-initialized output, (ii) the prior weight β in (4), and (iii) the decision temperature τ. It starts from the one-component estimate in raw principal coordinates, and Table 46 compares other starts.

N.1 Learning Objective

For a training dataset $t \in \mathcal{T}$ with candidate sources $\mathcal{S}_t$ from other cities, the listwise term matches the variant's distribution over sources to the transfer matrix's,

$$\mathcal{L}_{\text{list}} = \frac{1}{|\mathcal{T}|} \sum_{t} \mathrm{KL}\Big(\mathrm{softmax}_{s \in \mathcal{S}_t}\big(-P_{st}/\tau_P\big) \,\Big\|\, \mathrm{softmax}_{s \in \mathcal{S}_t}\big(-\widehat{P}_{s \to t}/\tau_d\big)\Big), \tag{11}$$

where the transfer entries and the estimates are standardized per batch, and τ_P and τ_d are temperatures (Table 10). A pairwise logistic term covers every source pair of a column whose transfer differs by more than a margin, and it grows when the estimate inverts the pair,

$$\mathcal{L}_{\text{pair}} = \frac{1}{|\mathcal{Q}|} \sum_{(a,b,t) \in \mathcal{Q}} \log\Big(1 + \exp\big(\big(\widehat{P}_{a \to t} - \widehat{P}_{b \to t}\big)/\eta\big)\Big), \quad \mathcal{Q} = \{(a,b,t) : P_{at} < P_{bt} - \xi\}, \tag{12}$$

where η is the standard deviation of the variant's estimates over the training pairs, recomputed at every step, and the fixed margin ξ is 0.05 standard deviations of the training transfer entries. The objective is the sum of (11) and (12). The number of epochs is chosen on validation cities held out from the training cities, and the final estimate averages five seeds.

On the DeepMIMO test cities, DiMTE (learned space) reaches 0.80 on LOS/NLOS identification and 0.76 on beam prediction, against 0.85 and 0.91 for DiMTE. On the Sionna test cities it reaches 0.73 and 0.70, against 0.78 and 0.79 (Tables 20 and 23). Supervision from measured transfer lifts its raw one-component start from 0.64 to 0.80 on LOS/NLOS identification (Table 46). Standardized coordinates and several components reach 0.85 without training on transfer outcomes, although k, M and τ are chosen on the transfer matrix of the training cities. The variant with its start and number of

components chosen on the training cities reaches 0.81 and 0.91 on the DeepMIMO test cities and 0.74 and 0.79 on the Sionna test cities. On LOS/NLOS identification it is behind DiMTE and recovers only 59% and 69% of the gap. On beam prediction the training cities chose the start at DiMTE, so the variant stays level with it.

Table 46: Controls for the learned variant on the training cities (city-grouped folds) against the measured transfer, per-target Spearman ρ. The shuffled control trains on the transfer matrix after relabeling the training datasets at random, for training and epoch selection alike, and is scored on the true matrix (mean of three shuffles). The standardized start uses the best learning rate of the grid. The rows with several components use the best M of their grid, and the exact start begins at DiMTE, with its standardized coordinates, components, shrunk variance and temperature. Training cities favor several components on LOS/NLOS identification and the exact start on beam prediction. Train columns use city-grouped folds of the training cities, and DeepMIMO columns score the same variants on the DeepMIMO test cities, where they were run. The last row is DiMTE without training, for comparison. Shaded rows use the paper's setting.

	LOS/NLOS		Beam prediction	
Training	Train	DeepMIMO	Train	DeepMIMO
DiMTE (learned space), true matrix, raw start	0.80	0.80	0.70	0.76
Shuffled matrix	0.29	0.47	0.31	0.39
Listwise loss only	0.80	–	0.69	–
Pairwise loss only	0.77	–	0.72	–
Standardized start	0.78	–	0.71	–
Several components per class ($M = 8$, $M = 4$)	0.83	0.81	0.69	–
Exact start at DiMTE	0.76	0.85	0.80	0.91
Raw one-component start, no training	–	0.64	–	0.57
DiMTE, no training (for comparison)	0.74	0.85	0.79	0.91

The learned variant learns from the transfer structure itself. The shuffled control of Table 46 trains the variant on a transfer matrix whose training datasets are relabeled at random, for training and epoch selection alike, and scores it on the true matrix. On the training-city folds the rank correlation falls from 0.80 to 0.29 on LOS/NLOS identification and from 0.70 to 0.31 on beam prediction. On the DeepMIMO test cities it falls to 0.47 and 0.39, below the untrained raw start (0.64 and 0.57). So the variant learns from the measured transfer, not from training as such. The listwise loss alone reaches 0.80 and 0.69, and the pairwise loss alone 0.77 and 0.72.

No start lifts the learned variant above DiMTE on the DeepMIMO test cities. A standardized start reaches 0.78 and 0.71 on the training-city folds at its best learning rate, and it was not scored on the test cities. Several components per class raise the correlation on the training cities on LOS/NLOS identification to 0.83, but on the DeepMIMO test cities that variant reaches 0.81, level with one component. A variant started exactly at DiMTE moves from 0.74 to 0.76 with training on the training cities. On the DeepMIMO test cities it ends at its starting value, 0.85 on LOS/NLOS identification and 0.91 on beam prediction. So the best start is DiMTE itself, which training leaves where it started. The learned variant keeps its original setting, the raw one-component start, which was fixed on the training cities before the mixture summary was introduced.

O Camera Traps, a Second Domain with Site-Specific Classes

DiMTE needs only a frozen encoder, class labels and a task metric, so it applies outside wireless. Camera traps are the closest analog of base stations that we found. Each camera location observes its own habitat, and the species it records differ between locations, as the beams and propagation states of base stations do. We use iWildCam2020-WILDS (Beery et al., 2020; Koh et al., 2021) and treat every camera location as one dataset and as its own group, the analog of a city. We keep the 32 most frequent species, drop empty frames, and keep the 65 locations with at least 400 images of these species. A seeded draw assigned 36 locations to training and 29 to test before any transfer outcome was computed. Within a location, camera sequences never cross the training, validation and test pools, so near-duplicate frames cannot leak between them. Each seed draws 200 training, 80 validation and 120 test images per location, resized to 224×224 pixels.

The measured transfer is the mean of two deployment models that share no representation. DINOv2 ViT-S/14 (Oquab et al., 2024) and a ResNet-50 pretrained on ImageNet (He et al., 2016), the

Table 47: Cross-domain source ranking on test groups, per-target Spearman ρ. *Two-learner mean* scores against the mean transfer of both fine-tuned deployment models, and *independent model* against the one that shares no representation with DiMTE. *Best baseline* is the strongest of all baselines. On MASSIVE DiMTE ties that baseline against the two-learner mean ($p = 0.820$ for our lead) and falls behind it on the independent model. Full protocols and baselines are in Appendices O and P.

		Two-learner mean		Independent model	
Benchmark	Targets	DiMTE	Best baseline	DiMTE	Best baseline
iWildCam	29 locations	0.87	0.81	0.82	0.79
MASSIVE	27 locales	0.66	0.67	0.51	0.56

architecture of the iWildCam2020-WILDS baseline, are fine-tuned end to end on each source location. Both use 224×224 inputs, horizontal flips, AdamW with weight decay 10^{-4}, a cosine schedule, 15 epochs, batches of 32, the epoch chosen on the source's validation images, and three seeds. The learning rate is 2×10^{-5} for the DINOv2 encoder and 10^{-4} for the ResNet-50 backbone, with 10^{-3} for both heads. On the test locations their in-site $1-$F1 is 0.30 and 0.32, and their cross-site $1-$F1 is 0.85 and 0.91, because most locations share few species. The two deployment models rank candidate sources at $\rho = 0.88$ against each other. An MLP on the frozen DINOv2 tokens reaches 0.30 in-site, as accurate as fine-tuned DINOv2, but it is not part of the measured transfer, because it shares the features of DiMTE. DiMTE and the embedding baselines embed images with the frozen DINOv2 `[CLS]` token, in the role of LWM. The baselines and the settings of DiMTE (learned space) are those of the wireless study. The hyperparameters were chosen on the training locations by the same two steps as for the wireless tasks (Appendices F and G), which gives $k = 64$, $\tau = 1$ and $M = 16$ components per class. The component grid is nearly flat on the training locations, and $M = 16$ has its largest ρ, 0.961 against 0.958 with one component.

On camera traps DiMTE leads, and shared species decide most of the transfer. On the 29 test locations DiMTE reaches $\rho = 0.87$ and DiMTE (learned space) 0.87, against 0.81 for the class-prior rule, the best of 43 baselines (Table 48). Every one-sided p of DiMTE against a baseline is below 0.001. The class prior is strong because the species that two locations share decide most of the transfer. DiMTE adds how those species look at each site. Symmetric label-aware distances average only over the classes that two locations share. Two locations without a shared species therefore get distance zero, and these distances rank sources in reverse. The set encoders trained on the transfer matrix of the 36 training locations generalize poorly, and the pairwise regressor falls to $\rho \approx 0$. DINOv2 is both the encoder of DiMTE and one of the two deployment models. We therefore also score every method against the fine-tuned ResNet-50 alone, which shares no representation with DiMTE. There DiMTE reaches 0.82 and DiMTE (learned space) 0.82, against 0.79 for the same class-prior rule ($p = 0.037$). Against fine-tuned DINOv2 alone DiMTE reaches 0.88, against 0.80 for that rule ($p < 0.001$). Figure 17 shows the most asymmetric pair of test locations.



(a) Location 186 (b) Location 13

Figure 17: The pair of camera test locations with the largest measured asymmetry, four photos from each location's training pool (iWildCam2020-WILDS, CDLA-Permissive 1.0). (a) Location 186 records 5 species. Shown, left to right and top to bottom, are the handsome francolin, the Gambian pouched rat, the black-fronted duiker and L'Hoest's monkey. (b) Location 13 records the black-fronted duiker in 96% of its images. A model trained on location 186 transfers to location 13 with a measured $1-$F1 of 0.35, while the reverse fails completely (1.00). DiMTE predicts 0.14 and 0.72.

Table 48: Source selection among the 29 camera-trap test locations of iWildCam2020-WILDS, against the mean transfer of DINOv2 and of an ImageNet ResNet-50, each fine-tuned end to end on 200 images of the source location. DiMTE and the embedding baselines use frozen DINOv2 embeddings. Learned methods were fit on the 36 training locations. Per-target Spearman ρ, gap recovered, and the selected source's mean $1-$F1 over 32 species. Best in bold, second best underlined. Shaded rows use the paper's setting. A correlation is written $0.78(1)$, where the bracketed digit is the standard error of the location means in the last place.

Distance	ρ	gap [metric]
Class prior		
Class prior (total variation)	$\underline{+0.81}_{(2)}$	<u>82%</u> [0.627]
Class prior (KL, directed)	$+0.74_{(3)}$	<u>82%</u> [0.627]
Raw pixels		
Label-free SW	$+0.39_{(5)}$	24% [0.804]
Label-aware centroid	$-0.63_{(4)}$	−40% [1.000]
Label-aware SW	$-0.63_{(4)}$	−40% [1.000]
Joint KL (directed), raw coordinates	$+0.30_{(5)}$	43% [0.746]
DiMTE, one component	$+0.76_{(3)}$	53% [0.715]
OTDD	$+0.20_{(5)}$	28% [0.794]
UMAP, 2-d	$-0.60_{(4)}$	−40% [1.000]
UMAP, 16-d	$-0.56_{(5)}$	−40% [1.000]
UMAP, 2-d, label-supervised	$-0.62_{(4)}$	−40% [1.000]
Task2Vec	$+0.50_{(4)}$	47% [0.734]
DINOv2 embeddings, no source model		
Label-aware centroid	$-0.63_{(4)}$	−40% [1.000]
Label-aware SW	$-0.61_{(4)}$	−40% [1.000]
Joint KL (directed), DINOv2 coordinates	$+0.56_{(4)}$	60% [0.695]
MMD, label-free	$+0.55_{(3)}$	52% [0.718]
MMD, label-aware	$+0.70_{(3)}$	37% [0.765]
CKA on class centroids	$+0.63_{(4)}$	17% [0.826]
Task2Vec	$+0.52_{(4)}$	44% [0.743]
Transferability estimators		
LEEP, closed-form source	$+0.61_{(5)}$	32% [0.780]
NCE, closed-form source	$+0.59_{(6)}$	30% [0.787]
OTCE, conditional entropy	$+0.51_{(6)}$	39% [0.758]
OTCE, transport + entropy	$+0.62_{(3)}$	50% [0.725]
LEEP, closed-form source, standardized	$+0.60_{(5)}$	32% [0.780]
NCE, closed-form source, standardized	$+0.59_{(6)}$	36% [0.767]
LEEP on our mixture classifier	$+0.60_{(4)}$	41% [0.753]
NCE on our mixture classifier	$+0.60_{(4)}$	40% [0.757]
OTCE, conditional entropy, standardized	$+0.54_{(6)}$	43% [0.747]
OTCE, transport + entropy, standardized	$+0.62_{(3)}$	52% [0.720]
LEEP, model trained per source	$+0.57_{(4)}$	44% [0.742]
NCE, model trained per source	$+0.52_{(5)}$	36% [0.767]
N-LEEP, model trained per source	$+0.41_{(4)}$	12% [0.843]
LogME, model trained per source	$+0.43_{(4)}$	36% [0.768]
H-score (regularized), model per source	$+0.31_{(5)}$	6% [0.862]
GBC, model trained per source	$+0.27_{(5)}$	9% [0.850]
TransRate, model trained per source	$+0.43_{(8)}$	20% [0.818]
PARC, model trained per source	$+0.34_{(4)}$	36% [0.767]
Learned without transfer supervision		
Self-supervised Deep Sets, DINOv2 embeddings	$+0.57_{(3)}$	46% [0.739]
Learned from the library transfer matrix		
Pairwise transfer predictor, raw descriptors	$+0.00_{(7)}$	31% [0.784]
Pairwise transfer predictor, DINOv2 descriptors	$+0.03_{(8)}$	13% [0.838]
Directed Deep Sets, raw pixels	$+0.61_{(2)}$	46% [0.738]
Directed Deep Sets, DINOv2 embeddings	$+0.62_{(2)}$	43% [0.747]
Directed Set Transformer, DINOv2 embeddings	$+0.65_{(3)}$	50% [0.726]
DiMTE (ours)		
Mixture classifier (default)	$\mathbf{+0.87}_{(2)}$	**86%** [0.617]
DiMTE (learned space)	$\mathbf{+0.87}_{(2)}$	78% [0.641]
Random source	–	0% [0.879]
Genie-aided source	–	100% [0.573]

P SOURCE-LANGUAGE SELECTION ON MASSIVE

The choice of a source language is a standard problem of cross-lingual transfer, because which language transfers best depends on the target (Lin et al., 2019). A new locale can then be trained on the labeled data of that language. The setting has what DiMTE needs, namely a frozen encoder, class labels and a task metric. The protocol follows the wireless study in four points: (i) datasets form groups, (ii) a seeded draw splits the groups into a training and a test set before any transfer outcome is computed, (iii) the hyperparameters are chosen on the training groups by the rule of Appendices F and G, and (iv) the measured transfer is the mean of two fine-tuned deployment models that share no representation. MASSIVE 1.1 (FitzGerald et al., 2023) gives the same virtual-assistant utterances, translated and localized by native speakers, in 52 locales, each labeled with one of 60 intents. A locale is one dataset and its language is its group, so the Chinese locales of mainland China and Taiwan never serve as each other's source. A seeded draw of the 51 languages assigned 25 locales to training and 27 to test. Each seed draws 1000 training utterances from a locale's training partition, 500 from its validation partition and 1000 from its test partition.

The measured transfer is the mean over two multilingual encoders fine-tuned end to end. Multilingual-e5-base (Wang et al., 2024) and multilingual BERT (Devlin et al., 2019) are fine-tuned with a linear head on mean-pooled tokens. Both use AdamW with a learning rate of 3×10^{-5} for the encoder and 10^{-3} for the head and a weight decay of 10^{-2}. They train with batches of 32 for 10 epochs, keep the epoch chosen on the source's validation utterances, and use three seeds. Multilingual-e5 is initialized from XLM-R (Conneau et al., 2020), and multilingual BERT shares neither weights nor tokenizer with it. On the test locales they reach an in-locale $1-$F1 of 0.23 and 0.37 and a cross-language $1-$F1 of 0.37 and 0.78. They rank candidate sources at $\rho = 0.65$ against each other, with seed reliability 0.75.

The frozen encoder is chosen on the training locales, and it has to keep languages apart. Transfer between locales is mostly a property of the source language. One order of sources explains most targets on the training locales (leave-one-out $\rho = 0.80$), and a source's own in-locale $1-$F1 predicts its mean transfer at 0.80. DiMTE therefore needs an encoder whose representations still differ between well and poorly covered languages. We compare the frozen backbones of the two deployment models and XLM-R (Conneau et al., 2020), from which multilingual-e5 is initialized, by the rank correlation of DiMTE with one component on the training locales. Multilingual BERT reaches 0.76, XLM-R 0.58 and multilingual-e5 0.32. Multilingual-e5 is trained contrastively on text pairs that include translations, which plausibly removes the differences between languages that decide transfer. We therefore embed utterances with frozen multilingual BERT, which is the backbone of one of the two deployment models, as LWM and DINOv2 are in the other checks. The hyperparameters chosen on the training locales are $k = 64$, $\tau = 1$ and $M = 2$.

The baselines add the standard methods of transfer-language selection. We keep the baselines of the wireless study, whose raw-feature versions read character n-gram TF-IDF (term frequency, inverse document frequency) vectors reduced to 256 dimensions by a truncated SVD fitted on the training locales. We add six URIEL distances between languages (Littell et al., 2017): (i) genetic, (ii) geographic, (iii) syntactic, (iv) phonological, (v) inventory, and (vi) featural. Each is computed from the lang2vec feature vectors of the two languages. We also score the word and subword overlap of the two training sets, and LangRank (Lin et al., 2019), a gradient-boosted ranker over all of these quantities trained on the measured transfer of the training locales.

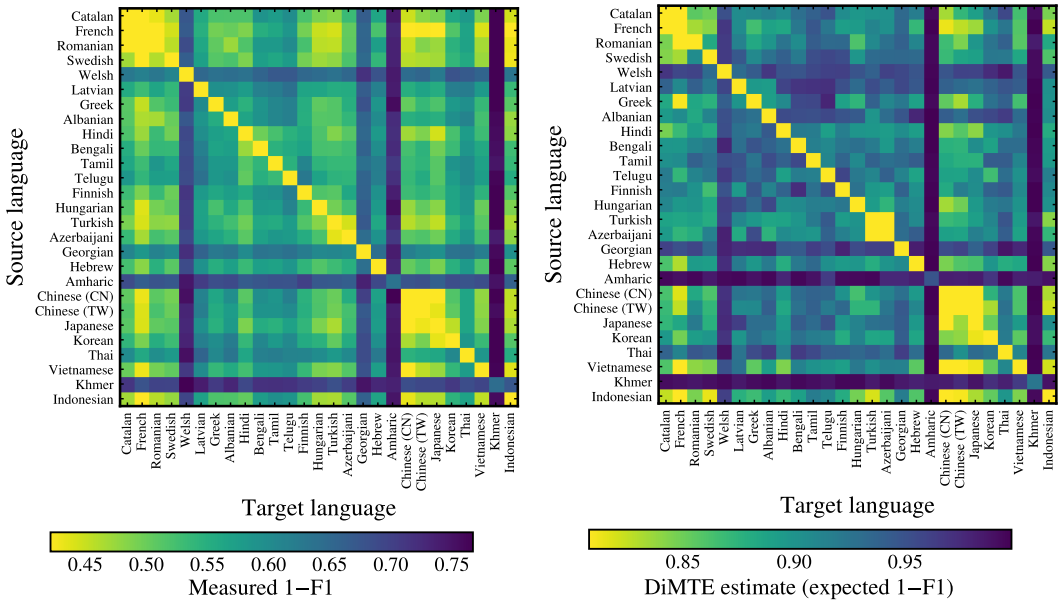
The frozen encoder ranks source languages far better than the transfer-language methods, and DiMTE only ties MMD in that space. On the 27 test locales DiMTE reaches $\rho = 0.66$ and DiMTE (learned space) 0.67 (Table 49). LangRank reaches 0.42, word overlap 0.42, and the best of the six URIEL distances, the syntactic distance, 0.28. Those features describe a pair of languages. A class-conditional summary of the source's own utterances also sees how well the encoder represents that language, which is what decides transfer here. The strongest of the 42 baselines is MMD, label-aware in the same fixed space at 0.67. DiMTE is within the noise of it ($p = 0.820$ for our lead), and DiMTE (learned space) reaches the same value. So on this benchmark the lead over the transfer-language features comes from the frozen encoder rather than from the directed estimate.

Multilingual BERT is also the backbone of one of the two deployment models, so we separate the two. Fine-tuned multilingual-e5 shares no representation with DiMTE. Against it alone, DiMTE reaches 0.51 and DiMTE (learned space) 0.57, against 0.56 for MMD, label-aware in the same space,

Table 49: Source-language selection on MASSIVE among 27 test locales of 26 languages, against the mean transfer of multilingual-e5 and of multilingual BERT, each fine-tuned end to end on 1000 utterances of the source locale. DiMTE and the embedding baselines use the frozen encoder chosen on the training locales, multilingual BERT. Learned methods and LangRank were fit on the 25 training locales. Per-target Spearman ρ, gap recovered, and the selected source's mean $1-$F1 over 60 intents. Best in bold, second best underlined. Shaded rows use the paper's setting. A correlation is written $0.78(1)$, where the bracketed digit is the standard error of the locale means in the last place.

Distance	ρ	gap [metric]
Class prior		
Class prior (total variation)	0.00(3)	12% [0.568]
Class prior (KL, directed)	+0.01(3)	18% [0.563]
Raw character n-grams		
Label-free SW	+0.14(8)	34% [0.550]
Label-aware centroid	+0.17(8)	44% [0.542]
Label-aware SW	+0.13(8)	29% [0.554]
DiMTE, one component	+0.34(6)	59% [0.530]
OTDD	+0.09(7)	10% [0.570]
UMAP, 2-d	+0.30(5)	27% [0.555]
UMAP, 16-d	+0.38(5)	52% [0.535]
UMAP, 2-d, label-supervised	+0.31(5)	29% [0.554]
mBERT embeddings, no source model		
Label-aware centroid	+0.64(4)	75% [0.516]
Label-aware SW	+0.58(5)	60% [0.529]
MMD, label-free	**+0.67**(5)	69% [0.521]
MMD, label-aware	**+0.67**(4)	75% [0.516]
CKA on class centroids	+0.41(4)	52% [0.535]
Transferability estimators		
LEEP, closed-form source	+0.59(5)	75% [0.516]
NCE, closed-form source	+0.54(5)	71% [0.520]
OTCE, conditional entropy	+0.40(3)	26% [0.556]
OTCE, transport + entropy	+0.47(3)	30% [0.553]
LEEP, model trained per source	<u>+0.66</u>(5)	54% [0.533]
NCE, model trained per source	+0.61(5)	37% [0.547]
N-LEEP, model trained per source	+0.56(6)	67% [0.523]
LogME, model trained per source	+0.62(6)	67% [0.523]
H-score (regularized), model per source	+0.58(5)	51% [0.536]
GBC, model trained per source	+0.21(3)	-84% [0.647]
TransRate, model trained per source	+0.47(5)	53% [0.534]
PARC, model trained per source	+0.58(6)	59% [0.529]
Learned without transfer supervision		
Self-supervised Deep Sets, mBERT embeddings	+0.40(5)	49% [0.538]
Learned from the library transfer matrix		
Pairwise transfer predictor, raw descriptors	+0.40(4)	51% [0.536]
Pairwise transfer predictor, mBERT descriptors	+0.40(5)	48% [0.539]
Directed Deep Sets, raw character n-grams	+0.31(5)	54% [0.533]
Directed Deep Sets, mBERT embeddings	+0.60(7)	**82%** [0.510]
Directed Set Transformer, mBERT embeddings	+0.57(6)	<u>80%</u> [0.512]
DiMTE (ours)		
Mixture classifier (default)	<u>+0.66</u>(5)	66% [0.523]
DiMTE (learned space)	**+0.67**(5)	79% [0.513]
Transfer-language selection		
URIEL genetic distance	+0.17(5)	-12% [0.587]
URIEL geographic distance	+0.20(4)	19% [0.562]
URIEL syntactic distance	+0.28(3)	50% [0.537]
URIEL phonological distance	+0.20(4)	-1% [0.578]
URIEL inventory distance	+0.11(3)	7% [0.572]
URIEL featural distance	+0.27(4)	39% [0.546]
Word overlap	+0.42(5)	67% [0.523]
Subword overlap	+0.32(4)	14% [0.566]
LangRank (trained on the training cities)	+0.42(4)	63% [0.526]
Random source	–	0% [0.578]
Genie-aided source	–	100% [0.496]

0.35 for LangRank and 0.30 for the best typology distance. Against fine-tuned multilingual BERT alone, DiMTE reaches 0.69 against 0.68 for label-free MMD ($p = 0.341$). So part of the lead over LangRank survives the control, while DiMTE falls behind the in-space MMD, label-aware on the independent deployment model. Two further points qualify the result. DiMTE (learned space), which reads the transfer matrix of the training locales, is the best of our formulations here, unlike on every wireless test set. The training rule also chose $M = 2$ components per class, while one component would have reached 0.72 on the test locales, above the chosen setting. The measured transfer is also less sharp than in wireless, with a seed reliability of 0.75 and an agreement of 0.65 between the two deployment models.



(a) Measured transfer

(b) DiMTE

Figure 18: The 27 MASSIVE test locales, ordered by language family. (a) Measured transfer, the mean 1−F1 of multilingual-e5 and multilingual BERT fine-tuned on the source locale. (b) DiMTE at its training-chosen settings, shown as the soft-confusion 1−F1 surrogate. Both show the Romance and Germanic block, the block of Chinese, Japanese and Korean, and the rows of Amharic and Khmer, whose data transfer poorly to every target.

Q A Third Task, Uplink Angle-of-Arrival Sectors

A third task tests the scores on a label defined by geometry. Uplink angle of arrival (UL-AoA) is a New Radio (NR) positioning measurement made at the base station, the gNB in 3GPP terms. We cast it as classification into eight sectors, UL-AoA sector classification. The label is the angle between the direction of the user equipment (UE) and the axis of the gNB's array, folded into $[0°, 180°]$ as a uniform linear array resolves it. The angle is computed from the ray-traced positions and binned into eight equal cones. The model observes the channel of an eight-antenna sub-array, as in beam prediction. The label comes from the user's position, so for NLOS users, whose strongest path arrives from a reflector, the channel carries it only indirectly. The datasets are those of the training cities and the DeepMIMO test cities of the other two tasks. The Sionna test cities carry no UL-AoA labels, so this task has no Sionna test.

On the training cities the measured transfer is at least as reliable as for the other tasks. The raw-CSI CNN reaches an in-site 1−F1 of 0.36 and a cross-site 1−F1 of 0.67, and fine-tuned LWM reaches 0.32 and 0.65. One training seed predicts the per-target order of the other seeds at 0.88 and 0.87, and the two deployment models agree with each other at 0.88. The MLP on frozen tokens is weaker, with an in-site 1−F1 of 0.45. The measured transfer is again the two-learner mean of these two deployment models.

Table 50: UL-AoA sector classification on the training cities (city-grouped folds) against the measured transfer. Per-target Spearman ρ, share of the random-to-genie gap recovered, and the correlation between predicted and measured transfer asymmetry (directional ρ). Every LWM-space method uses standardized coordinates. DiMTE uses the subspace size and temperature chosen on these cities ($k = 64$, $\tau = 0.3$) and the number of components chosen there (Table 14). DiMTE (learned space) starts from those standardized $k = 64$ coordinates with one component per class, while on LOS/NLOS identification and beam prediction it starts from raw coordinates. Best in bold, second best underlined. Shaded rows use the paper's setting.

Distance	ρ	gap	directional ρ
Class prior			
Class prior (total variation)	+0.43	31%	+0.00
Class prior (KL, directed)	+0.47	37%	+0.27
Raw channels			
Label-free SW	+0.55	53%	+0.00
Label-aware centroid	+0.28	7%	+0.00
Label-aware SW	+0.63	62%	+0.00
Joint KL (directed), raw coordinates	+0.35	57%	+0.34
DiMTE, one component	+0.50	51%	+0.36
OTDD	+0.50	57%	+0.00
UMAP, 2-d	+0.40	38%	+0.00
UMAP, 16-d	+0.48	58%	+0.00
UMAP, 2-d, label-supervised	+0.44	55%	+0.00
Task2Vec	+0.59	70%	+0.00
LWM embeddings, no source model			
Label-aware centroid	+0.43	61%	+0.00
Label-aware SW	+0.47	62%	+0.00
Joint KL (directed), LWM coordinates	+0.68	69%	−0.18
MMD, label-free	+0.13	32%	+0.00
MMD, label-aware	+0.39	52%	+0.00
CKA on class centroids	+0.06	27%	+0.00
Task2Vec	+0.65	72%	+0.00
Transferability estimators			
LEEP, closed-form source	+0.50	61%	+0.57
NCE, closed-form source	+0.58	56%	+0.52
OTCE, conditional entropy	+0.54	64%	+0.27
OTCE, transport + entropy	+0.25	60%	+0.27
LEEP, model trained per source	+0.60	76%	+0.43
NCE, model trained per source	+0.63	70%	+0.44
N-LEEP, model trained per source	+0.55	51%	+0.46
LogME, model trained per source	+0.55	60%	+0.33
H-score (regularized), model per source	+0.56	58%	+0.46
GBC, model trained per source	+0.45	56%	+0.08
TransRate, model trained per source	+0.55	50%	−0.14
PARC, model trained per source	+0.38	53%	+0.40
Learned without transfer supervision			
Self-supervised Deep Sets, LWM embeddings	+0.25	72%	+0.00
Learned from the library transfer matrix			
Directed Deep Sets, LWM embeddings	<u>+0.81</u>	81%	−0.04
Directed Set Transformer, LWM embeddings	+0.64	64%	−0.24
DiMTE (ours)			
Mixture classifier (default)	**+0.88**	**91%**	+0.76
DiMTE (learned space)	+0.79	<u>84%</u>	+0.72
Variant (Appendix G)			
DiMTE, one component per class	+0.77	82%	+0.78

The mixture summary ranks UL-AoA sources best on the training cities. Table 50 scores every method. With the settings chosen on these cities, $k = 64$, $\tau = 0.3$ and $M = 128$, DiMTE reaches $\rho = 0.88$, against 0.77 with one component per class, and DiMTE (learned space) reaches 0.79. The best score that reads neither a trained model nor the transfer matrix, directed joint KL on LWM embeddings, reaches 0.68. The strongest baseline, Directed Deep Sets, LWM embeddings, reaches 0.81, yet its predicted asymmetry correlates with the measured asymmetry only at -0.04, against +0.76 for DiMTE and +0.72 for DiMTE (learned space). Directed Deep Sets reaches only 0.58 on the training cities of LOS/NLOS identification (Table 19), so its advantage does not carry over between tasks.

On the DeepMIMO test cities, DiMTE and its learned variant both lead. DiMTE, with every setting fixed on the training cities, reaches $\rho = 0.94$ on the 84 targets of these cities, against 0.87 with one component per class. DiMTE (learned space), which starts from standardized coordinates on this task (Table 50), reaches 0.85 (Table 51). The strongest of the 43 baselines, NCE on our mixture classifier, reaches 0.79. Every one-sided p against a baseline is below 0.001 for DiMTE and at most 0.003 for DiMTE (learned space).

R History and Corrections of the Protocol

This appendix records how the evaluation protocol changed during the study, and why. An earlier version of this study evaluated dataset distances with a dataset-level split, a single-seed transfer matrix, and correlations against the symmetrized matrix $\frac{1}{2}(\mathbf{P} + \mathbf{P}^\top)$. Each of these choices made the evaluation less reliable, and two of them favored the methods under test. Base stations of one city appeared on both sides of the split, so a learned embedding was scored on environments it had been fit to. A single-seed matrix has a replicate discrepancy comparable to the differences between methods, so rankings among methods changed from seed to seed. The symmetrized matrix also removes the component of $\mathbf{P}$ that a symmetric score cannot represent. It raised the correlation of every symmetric score by about 0.2 on the same data. Under the corrected protocol of Section 5, the conclusions of that version did not hold. The learned refinement of that version was matched by a control trained on a randomly permuted transfer matrix, and its estimate did not beat a raw-channel distance. The present method was developed against the corrected protocol, with every setting selected by training-city results.

The DeepMIMO test set was scored before the final design existed, so a separate test set was added. The design went through four stages: (i) the one-component estimate in raw principal coordinates and DiMTE (learned space) were scored against the matrix of the MLP on frozen tokens, described below, (ii) the two-learner mean defined in Section 5.2 replaced that matrix, (iii) standardized coordinates were introduced, and (iv) the mixture summary followed. Every setting in stages (iii) and (iv) was chosen by training-city results, but the DeepMIMO test results of the earlier stages were known, so they may have shaped which ideas we tried. Two checks bound this risk. A choice of M on half of the DeepMIMO test cities, scored on the other half, gives the same conclusions (Appendix K). The Sionna test set of Section 6.1 was generated only after every choice was fixed, in cities outside DeepMIMO and with another ray tracer, and no setting of DiMTE was chosen on it. The only choices made on it favor baselines (Table 8).

Against the earlier matrix of the MLP on frozen LWM tokens, DiMTE and its learned variant both led every baseline of that version. The version of this study that preceded Section 5.2 measured transfer with an MLP on frozen LWM channel tokens. That deployment model is weak and shares its features with DiMTE, so its numbers are not comparable with the rest of the paper. Against that matrix on the DeepMIMO test cities, DiMTE, then in raw principal coordinates with one component per class, reached 0.80 on LOS/NLOS identification and 0.81 on beam prediction, which then read the full-array channel. DiMTE (learned space) reached 0.84 and 0.85, so there it ranked above DiMTE, the opposite of the current result with standardized coordinates and several components. DiMTE led the best baseline of each task, LEEP, closed-form source classifier (LWM) and OTDD on raw channels, by +0.08 and +0.05, and DiMTE (learned space) led them by +0.12 and +0.09. The loss ablations of DiMTE (learned space) were repeated against the two-learner mean (Table 46).

Table 51: UL-AoA sector classification on the DeepMIMO test cities against the measured transfer, with every setting chosen on the training cities. Per-target Spearman ρ, gap recovered, and the selected source's mean $1-\mathrm{F1}$. Best in bold, second best underlined. Shaded rows use the paper's setting.

Distance	ρ	gap [metric]
Class prior		
Class prior (total variation)	+0.62	53% [0.413]
Class prior (KL, directed)	+0.66	60% [0.397]
Raw channels		
Label-free SW	+0.63	67% [0.380]
Label-aware centroid	+0.50	35% [0.459]
Label-aware SW	+0.75	71% [0.369]
Joint KL (directed), raw coordinates	+0.41	61% [0.394]
DiMTE, one component	+0.64	46% [0.432]
OTDD	+0.61	59% [0.399]
UMAP, 2-d	+0.57	59% [0.399]
UMAP, 16-d	+0.64	68% [0.376]
UMAP, 2-d, label-supervised	+0.57	58% [0.403]
Task2Vec	+0.69	80% [0.347]
LWM embeddings, no source model		
Label-aware centroid	+0.53	68% [0.378]
Label-aware SW	+0.59	67% [0.379]
Joint KL (directed), LWM coordinates	+0.68	77% [0.355]
MMD, label-free	+0.20	31% [0.468]
MMD, label-aware	+0.54	66% [0.382]
CKA on class centroids	+0.10	17% [0.503]
Task2Vec	+0.68	75% [0.360]
Transferability estimators		
LEEP, closed-form source	−0.10	−47% [0.661]
NCE, closed-form source	+0.44	43% [0.439]
OTCE, conditional entropy	+0.51	46% [0.431]
OTCE, transport + entropy	+0.38	53% [0.413]
LEEP, closed-form source, standardized	+0.56	76% [0.356]
NCE, closed-form source, standardized	+0.62	65% [0.384]
LEEP on our mixture classifier	+0.79	82% [0.343]
NCE on our mixture classifier	+0.79	80% [0.346]
OTCE, conditional entropy, standardized	+0.68	75% [0.360]
OTCE, transport + entropy, standardized	+0.55	74% [0.363]
LEEP, model trained per source	+0.69	81% [0.345]
NCE, model trained per source	+0.72	73% [0.365]
N-LEEP, model trained per source	+0.65	60% [0.397]
LogME, model trained per source	+0.62	70% [0.372]
H-score (regularized), model per source	+0.62	65% [0.384]
GBC, model trained per source	+0.53	57% [0.404]
TransRate, model trained per source	+0.61	72% [0.367]
PARC, model trained per source	+0.64	77% [0.354]
Learned without transfer supervision		
Self-supervised Deep Sets, LWM embeddings	+0.27	65% [0.385]
Learned from the library transfer matrix		
Pairwise transfer predictor, raw descriptors	+0.58	49% [0.423]
Pairwise transfer predictor, LWM descriptors	+0.61	76% [0.357]
Directed Deep Sets, raw channels	+0.68	17% [0.502]
Directed Deep Sets, LWM embeddings	+0.72	71% [0.368]
Directed Set Transformer, LWM embeddings	+0.58	46% [0.431]
DiMTE (ours)		
Mixture classifier (default)	**+0.94**	**92%** [0.318]
DiMTE (learned space)	<u>+0.85</u>	<u>85%</u> [0.334]
Random source	–	0% [0.545]
Genie-aided source	–	100% [0.298]

S TERMS USED IN THIS PAPER

This appendix defines, in plain words, the wireless and method terms of the main text and the appendices.

Site. A base station and the users it serves in one area. Each site gives one dataset.

Sample. The narrowband radio channel between the 32 antennas of a site and one user, the model input.

LOS/NLOS identification. Binary classification of whether a user's direct path to the base station is clear (line of sight, LOS) or blocked by buildings (non-line of sight, NLOS). The loss is $1-$F1 with class-weighted F1.

Beam prediction. Spatial-domain downlink beam prediction, a 32-class choice of the transmit direction, or beam, that serves a user best, among 32 discrete Fourier transform (DFT) beams.

Codebook. The fixed set of beams that a base station can use. Here it holds 32 discrete Fourier transform (DFT) beams, each pointing in one direction.

L1-RSRP difference. The loss of beam prediction in dB, namely the power a user loses when the predicted beam serves it instead of the genie-aided one. L1-RSRP is the Layer-1 reference-signal received power that a user measures for each beam.

Genie-aided. Chosen with full knowledge of the outcome, that is, the best beam or the best source in hindsight.

Top-1/1 accuracy. The share of users whose best beam is predicted.

Channel state information (CSI). The measured complex channel. The beam-prediction model reads the CSI of an eight-antenna sub-array, eight of the 32 antennas.

gNB and UE. The 3GPP names of the base station (gNB) and of the user device (UE).

Ray tracing. Simulating radio propagation, path by path, through a 3D city model.

DeepMIMO. A public collection of ray-traced city scenarios (Alkhateeb, 2019). It gives the training cities and the DeepMIMO test cities.

Sionna RT. An open-source ray tracer (Aït Aoudia et al., 2025). We used it to build the Sionna test cities after the method was frozen.

Training, DeepMIMO test and Sionna test cities. The three city sets. The training cities choose k, M and τ and train the set encoders, the DeepMIMO test cities were scored during design, and the Sionna test cities give the confirmatory results.

Library. The labeled datasets a practitioner already holds, the candidate sources for a new target.

Transfer profile. The loss that a model trained on each library dataset would reach on the target.

Deployment model. The model that a specified training procedure produces from a source and that would be deployed at the target. The measured transfer uses fine-tuned LWM and a CNN trained on raw CSI.

Measured transfer. The target loss of deployment models actually trained on each source, averaged over training seeds and over the two deployment models.

Two-learner mean. The measured transfer averaged over fine-tuned LWM and the raw-CSI CNN, two deployment models that share no representation.

Transfer matrix. The measured transfer between every pair of datasets, here of the training cities.

Proxy and source summary. The cheap classifier that DiMTE builds from a source, and what it stores for it, namely M prototypes per class, one tied variance and the class priors.

Readout. The deployment loss under which DiMTE evaluates the proxy on the target samples, such as the L1-RSRP difference or $1-$F1. The Top-1 readout uses the Top-1 error, which needs only the beam labels.

k-NN and Parzen rules. Two other source classifiers inside DiMTE. The k-NN rule averages the labels of the nearest source samples, and the Parzen rule sums a Gaussian kernel over the source samples of each class.

Hyperparameters k, M and τ. The number of standardized principal coordinates, the number of prototypes per class and the softmax temperature, chosen once per task on the training cities.

Training-free score. A baseline that trains nothing per source, such as a dataset distance or a class-prior divergence.

Set encoder. A baseline that embeds a whole dataset as a set and, in its supervised form, learns to predict transfer from the transfer matrix.

Deep Sets and Set Transformer. Neural networks that embed a dataset as an unordered set of samples (Zaheer et al., 2017; Lee et al., 2019). Our directed versions learn to predict transfer from the transfer matrix of the training cities.

Learned variant. DiMTE (learned space), which learns the embedding space of DiMTE from the transfer matrix of the training cities (Appendix N).

Model-aware estimator. A transferability estimator, such as LEEP or NCE, that reads a deployment model trained on every source.

Joint KL. A directed KL divergence between the joint distributions of features and labels of two datasets.

Class-prior KL. A directed KL divergence between the label distributions of two datasets.

LEEP and NCE. Transferability estimators that score a source from its classifier's predictions on the target samples. *On our mixture* means that they read the proxy of DiMTE instead of a trained model.

OTCE, JC-NCE and WDJE. Transferability scores built on optimal transport. OTCE and JC-NCE score the conditional entropy of the target labels given source labels matched by optimal transport (Tan et al., 2021; 2024). WDJE bounds the target risk by the source risk plus Wasserstein distances between the features and between the labels of the two datasets (Zhan et al., 2026).

OTDD. The optimal-transport dataset distance, which compares two labeled datasets.

UL-AoA. Uplink angle-of-arrival sector classification, the third wireless task (Appendix Q).

Sub-array sweep. A label-free guess of a user's best beam, which scores each of the 32 beams on the eight-antenna input channel.

Pseudo-label. A label predicted without measurement, from a classifier trained on the library for LOS/NLOS identification and from the sub-array sweep for beam prediction (Appendix L).

Camera traps. Wildlife cameras of iWildCam. Each camera location acts as a site (Appendix O).

MASSIVE. A multilingual intent-classification benchmark. Each language acts as a source (Appendix P).

ρ**.** The Spearman correlation between estimated and measured transfer over the candidate sources of one target, averaged over targets.

Gap recovered. The share of the loss difference between a random source and the genie-aided source that the top-ranked source recovers, pooled over targets.

Regret. The loss of the top-ranked source minus the loss of the genie-aided source.

Isotonic map. A monotone function from a score to the deployment metric, fitted by isotonic regression on the cross-city pairs of the training cities. It changes no ranking.

Support score. The share of target samples whose Mahalanobis distance to the nearest source component exceeds the 95th percentile of the source's own samples (Appendix E).

Facility location. A greedy rule that picks R prototype sites one at a time, each time the one that most lowers the summed score of every site to its best prototype (Appendix K).