Transfer Learning Through Conditional Quantile Matching

Yikun Zhang[1]

Joint work with *Steven Wilkins-Reeves*[2], *Wesley Lee*[2], and *Aude Hofleitner*[2]

[1]Department of Statistics, University of Washington
[2]Central Applied Science, Meta

New Horizons on Model Transportability and Data Integration
June 23, 2026

UNIVERSITY of WASHINGTON

Central Applied Science

Meta

Business Question: How do Meta's family of apps perform compared to an app developed by a competing company?

Business Question: How do Meta's family of apps perform compared to an app developed by a competing company?

- **Metric:** Country-level monthly active user (MAU) statistics for an app.
- **Limitation:** Complete MAU data for Meta's apps but limited MAU data for the target app.

Business Question: How do Meta's family of apps perform compared to an app developed by a competing company?

- **Metric:** Country-level monthly active user (MAU) statistics for an app.
- **Limitation:** Complete MAU data for Meta's apps but limited MAU data for the target app.

Objective: Predict the country-level MAU of a target app using data from Meta's family of apps.

Regression Task: Predict the country-level MAU $Y^{(0)} \in \mathcal{Y}$ for a target app from country-specific covariates $X^{(0)} \in \mathcal{X}$.

1. **Source-Domain Data:** $\mathcal{D}_S^{(k)} = \left\{ \left(X_i^{(k)}, Y_i^{(k)} \right) \right\}_{i=1}^{n_k} \sim P^{(k)}$ for $k = 1, \ldots, K$.
2. **Target-Domain Data:** $\mathcal{D}_T = \left\{ \left(X_i^{(0)}, Y_i^{(0)} \right) \right\}_{i=1}^{n_0} \sim P^{(0)}$, where $n_0 \ll n_k$ for $k = 1, \ldots, K$.

Regression Task: Predict the country-level MAU $Y^{(0)} \in \mathcal{Y}$ for a target app from country-specific covariates $X^{(0)} \in \mathcal{X}$.

1. **Source-Domain Data:** $\mathcal{D}_S^{(k)} = \left\{ \left(X_i^{(k)}, Y_i^{(k)} \right) \right\}_{i=1}^{n_k} \sim P^{(k)}$ for $k = 1, \ldots, K$.
2. **Target-Domain Data:** $\mathcal{D}_T = \left\{ \left(X_i^{(0)}, Y_i^{(0)} \right) \right\}_{i=1}^{n_0} \sim P^{(0)}$, where $n_0 \ll n_k$ for $k = 1, \ldots, K$.

Covariate Shift (Shimodaira, 2000):
$P^{(k)}(Y|X) = P^{(0)}(Y|X)$ but $P^{(k)}(X) \neq P^{(0)}(X)$.

Label Shift (Saerens et al., 2002):
$P^{(k)}(X|Y) = P^{(0)}(X|Y)$ but $P^{(k)}(Y) \neq P^{(0)}(Y)$.

General Target Shift:

$$P^{(k)}(X) \neq P^{(0)}(X) \quad \text{and} \quad P^{(k)}(Y|X) \neq P^{(0)}(Y|X) \quad \text{for } k = 1, \ldots, K.$$

Proposed Transfer Learning Framework (TLCQM)

Key Idea: Jointly calibrate source distributions for data augmentation in the target domain (Zhang et al., 2026).

Key Idea: Jointly calibrate source distributions for data augmentation in the target domain (Zhang et al., 2026).

1 Learn source distributions $\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \ldots, K$. $\longrightarrow$

1. Learn a generative model $\widehat{P}^{(k)}(\cdot|x)$ using $\left\{ \left(X_i^{(k)}, Y_i^{(k)} \right) \right\}_{i=1}^{n_k}$ for each source domain k.

Key Idea: Jointly calibrate source distributions for data augmentation in the target domain (Zhang et al., 2026).

1 Learn source distributions $\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \ldots, K$. $\longrightarrow$

1. Learn a generative model $\widehat{P}^{(k)}(\cdot|x)$ using $\left\{\left(X_i^{(k)}, Y_i^{(k)}\right)\right\}_{i=1}^{n_k}$ for each source domain k.
   - Model $P^{(k)}(\cdot|x) = g^{(k)}(x, \cdot)_{\#}P_\eta$ and estimate $\widehat{g}^{(k)}$ via engression (Shen and Meinshausen, 2025):

$$\widehat{g}^{(k)} \in \underset{g \in \mathcal{G}}{\arg\min} \frac{1}{n_k} \sum_{i=1}^{n_k} \left[\frac{1}{m} \sum_{j=1}^{m} \left| Y_i^{(k)} - g(X_i^{(k)}, \eta_{ij}) \right| - \frac{1}{2m(m-1)} \sum_{j=1}^{m} \sum_{j'=1}^{m} \left| g(X_i^{(k)}, \eta_{ij}) - g(X_i^{(k)}, \eta_{ij'}) \right| \right],$$

   where P_η is a prespecified noise distribution and $\mathcal{G}$ is a neural network class.

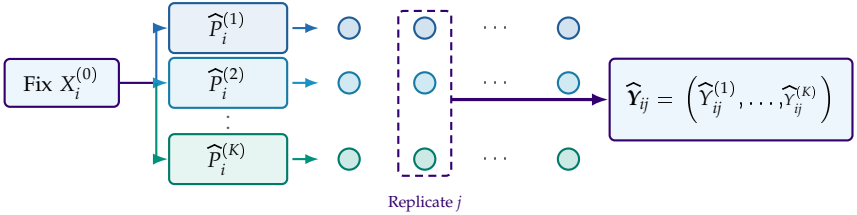
1 Learn source distributions
$\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \ldots, K$.

→

2 Generate at target covariates
$\widehat{Y}_{ij}^{(k)} \sim \widehat{P}^{(k)}(\cdot|X_i^{(0)})$ for $j = 1, \ldots, M$.

→

2. For each target covariate $X_i^{(0)}$, independently draw M synthetic response vectors:



$$\left\{\left(X_i^{(0)}, \widehat{\mathbf{Y}}_{ij}\right) : i = 1, \ldots, n_0,\ j = 1, \ldots, M\right\}, \quad \widehat{\mathbf{Y}}_{ij} = \left(\widehat{Y}_{ij}^{(1)}, \ldots, \widehat{Y}_{ij}^{(K)}\right), \quad \widehat{Y}_{ij}^{(k)} \sim \widehat{P}^{(k)}\left(\cdot|X_i^{(0)}\right).$$

1 Learn source distributions
$\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \ldots, K$.

→

2 Generate at target covariates
$\widehat{Y}_{ij}^{(k)} \sim \widehat{P}^{(k)}(\cdot|X_i^{(0)})$ for $j = 1, \ldots, M$.

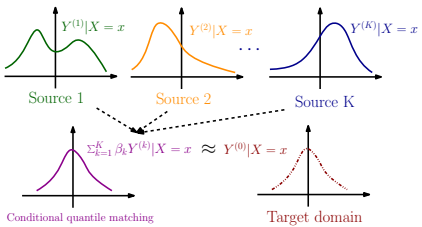
→

3 Conditional quantile calibration
$\widehat{\beta}_0 + \sum_{k=1}^{K} \widehat{\beta}_k \widehat{Y}^{(k)}$ matches the target distribution.

3. Compute the quantile matching estimator (Sgouropoulos et al., 2015):

$$\widehat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta} \in \mathbb{R}^{K+1}}{\arg\min} \sum_{i=1}^{n_0} \sum_{j=1}^{M} \left[Y_{(i)}^{(0)} - \left(\boldsymbol{\beta}^T \widehat{\boldsymbol{V}} \right)_{((i-1)M+j)} \right]^2.$$

- $\widehat{\boldsymbol{V}}_{ij} = \left(1, \widehat{\boldsymbol{Y}}_{ij}\right)^T$, while $Y_{(1)}^{(0)} \leq \cdots \leq Y_{(n_0)}^{(0)}$ and $\left(\boldsymbol{\beta}^T \widehat{\boldsymbol{V}} \right)_{(1)} \leq \cdots \leq \left(\boldsymbol{\beta}^T \widehat{\boldsymbol{V}} \right)_{(n_0 M)}$ are order statistics, respectively.



Proposed Transfer Learning Framework (TLCQM)

1 Learn source distributions
$\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \ldots, K$.

→

2 Generate at target covariates
$\widehat{Y}_{ij}^{(k)} \sim \widehat{P}^{(k)}(\cdot|X_i^{(0)})$ for $j = 1, \ldots, M$.

→

3 Conditional quantile calibration
$\widehat{\beta}_0 + \sum_{k=1}^{K} \widehat{\beta}_k \widehat{Y}^{(k)}$ matches the target distribution.

→

4 Augment and fit on $\mathcal{D}_T \cup \mathcal{D}_A$
XGBoost, kernel ridge regression (KRR), neural network (NN),...

4. Augment the target domain with synthetic, target-calibrated labels:

$$\mathcal{D}_A = \bigcup_{k=1}^{K} \left\{ \left(X_i^{(k)}, \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{V}}_i^{(k)} \right) \right\}_{i=1}^{n_k} \quad \text{with} \quad \widehat{\boldsymbol{V}}_i^{(k)} = \left(1, \widehat{Y}_i^{(1,k)}, \ldots, \widehat{Y}_i^{(K,k)}\right)^T \in \mathbb{R}^{K+1},$$

where $\widehat{Y}_i^{(j,k)}$ is a prediction of $Y^{(j)}$ at $X_i^{(k)}$, such as $\widehat{\mathbb{E}}(Y^{(j)}|X_i^{(k)})$.

5. (Optional) Estimate $w_k(x) = \frac{dP_X^{(0)}(x)}{dP_X^{(k)}(x)}$ for $k = 1, \ldots, K$ to correct for covariate shift.

Theory: Excess Risk Decomposition

- $R(f) := \mathbb{E}_{P^{(0)}}\left[\ell\left(Y^{(0)}, f(X^{(0)})\right)\right]$ and $f^{(0)} \in \arg\min_{f\in\mathcal{F}} R(f)$ under a loss function ℓ.
- $\mathrm{Rad}_n(\mathcal{F})$ is the Rademacher complexity of $\mathcal{F}$.

Target-Only: $\widehat{f}^{(0)} = \arg\min_{f\in\mathcal{F}} \frac{1}{n_0}\sum_{i=1}^{n_0} \ell\left(Y_i^{(0)}, f(X_i^{(0)})\right)$ satisfies

$$R(\widehat{f}^{(0)}) - R(f^{(0)}) \lesssim \mathrm{Rad}_{n_0}(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{n_0}} \quad \text{w. p. } \geq 1-\delta.$$

- $R(f) := \mathbb{E}_{P^{(0)}}\left[\ell\left(Y^{(0)}, f(X^{(0)})\right)\right]$ and $f^{(0)} \in \arg\min_{f\in\mathcal{F}} R(f)$ under a loss function ℓ.
- $\mathrm{Rad}_n(\mathcal{F})$ is the Rademacher complexity of $\mathcal{F}$.

Target-Only: $\widehat{f}^{(0)} = \arg\min_{f\in\mathcal{F}} \frac{1}{n_0}\sum_{i=1}^{n_0} \ell\left(Y_i^{(0)}, f(X_i^{(0)})\right)$ satisfies

$$R(\widehat{f}^{(0)}) - R(f^{(0)}) \lesssim \mathrm{Rad}_{n_0}(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{n_0}} \quad \text{w. p. } \geq 1-\delta.$$

TLCQM: $\widehat{f}^{(0,tl)} = \arg\min_{f\in\mathcal{F}} \frac{1}{N}\left\{\sum_{i=1}^{n_0} \ell\left(Y_i^{(0)}, f(X_i^{(0)})\right) + \sum_{k=1}^{K}\sum_{i=1}^{n_k} \widehat{w}_k(X_i^{(k)}) \cdot \ell\left(\widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{V}}_i^{(k)}, f(X_i^{(k)})\right)\right\}$ satisfies

$$R(\widehat{f}^{(0,tl)}) - R(f^{(0)}) \lesssim \mathrm{Rad}_N(\mathcal{F}) + \sqrt{\frac{K\log(1/\delta)}{N}} + \frac{1}{N}\sum_{k=1}^{K} \|\widehat{w}_k - w_k\|_1 + \inf_{\boldsymbol{\beta}_*\in\mathcal{B}} \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_*\|_1$$
$$+ \sum_{k=1}^{K} \|\widehat{g}^{(k)} - g^{(k)}\|_\infty + \inf_{\boldsymbol{\beta}_*\in\mathcal{B}} \left(\int_0^1 \left[Q_{Y^{(0)}}(\alpha) - Q_{\boldsymbol{\beta}_*^T \boldsymbol{V}}(\alpha)\right]^2 d\alpha\right)^{1/2} \quad \text{w. p. } \geq 1-\delta,$$

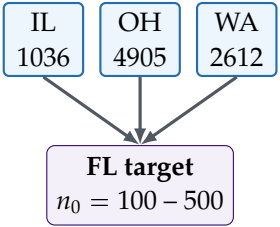
where $N = \sum_{k=0}^{K} n_k$ and $\mathcal{B} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{K+1}} \int_0^1 \left[Q_{Y^{(0)}}(\alpha) - Q_{\boldsymbol{\beta}^T \boldsymbol{V}}(\alpha)\right]^2 d\alpha$.

$$R(\widehat{f}^{(0,tl)}) - R(f^{(0)}) \lesssim \underbrace{\mathrm{Rad}_N(\mathcal{F}) + \sqrt{\frac{K\log(1/\delta)}{N}}}_{\text{Standard generalization error}} + \underbrace{\frac{1}{N}\sum_{k=1}^{K}\|\widehat{w}_k - w_k\|_1}_{\text{Importance weight error}} + \underbrace{\sum_{k=1}^{K}\|\widehat{g}^{(k)} - g^{(k)}\|_\infty}_{\text{Distributional learning error}}$$

$$+ \underbrace{\inf_{\boldsymbol{\beta}_* \in \mathcal{B}} \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_*\|_1}_{\text{Quantile matching error}} + \underbrace{\inf_{\boldsymbol{\beta}_* \in \mathcal{B}} \left(\int_0^1 [Q_{Y^{(0)}}(\alpha) - Q_{\boldsymbol{\beta}_*^T \boldsymbol{V}}(\alpha)]^2 \, d\alpha\right)^{1/2}}_{\text{Transfer bias}} \quad \text{w. p. } \geq 1-\delta.$$

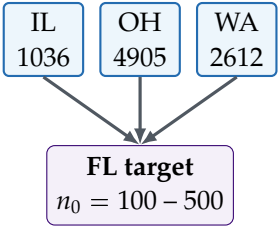
1. $N = \sum_{k=0}^{K} n_k \gg n_0$ reduces the generalization/complexity error.
2. $\inf_{\boldsymbol{\beta}_* \in \mathcal{B}} \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_*\|_1 = O\left(\sqrt{K\sum_{k=1}^{K}\|\widehat{g}^{(k)} - g^{(k)}\|_\infty}\right) + O_P\left(\sqrt{\frac{K\log\log n_0}{n_0}} + \sqrt{K}\left[\frac{\log\log n_0}{n_0} \cdot \inf_{\boldsymbol{\beta}_* \in \mathcal{B}} \int_0^1 \{Q_{Y^{(0)}}(\alpha) - Q_{\boldsymbol{\beta}_*^T \boldsymbol{V}}(\alpha)\}^2 \, d\alpha\right]^{1/4}\right)$.
3. Transfer bias vanishes when $P^{(0)}(Y|X)$ lies in the convex hull of $\{P^{(k)}(Y|X)\}_{k=1}^{K}$.

Goal: Predict apartment rental prices in FL using data from IL, OH, and WA.



500 Monte Carlo replications.

Goal: Predict apartment rental prices in FL using data from IL, OH, and WA.

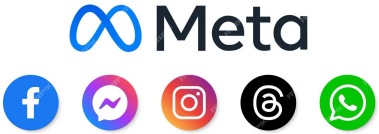


500 Monte Carlo replications.

MSE reduction versus target-only training

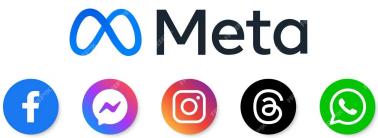
n_0	100	200	300	500
XGBoost	7.2%	1.6%	−2.3%	−6.8%
KRR	**94.2%**	**91.4%**	**88.4%**	**82.5%**
NN	**99.4%**	**99.4%**	**96.9%**	**88.0%**

Result: MSE reductions are larger for more flexible learners (NN and KRR) and generally decrease as n_0 increases.



Objective: Predict country-level MAU for a held-out app at Meta.

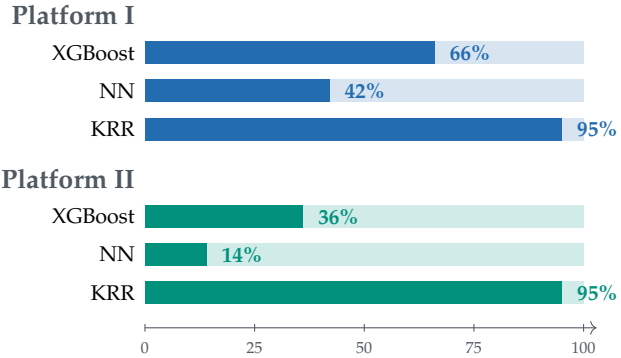
- Four source apps with ≈ 230 observations per app across two device platforms.



Objective: Predict country-level MAU for a held-out app at Meta.

- Four source apps with ≈ 230 observations per app across two device platforms.

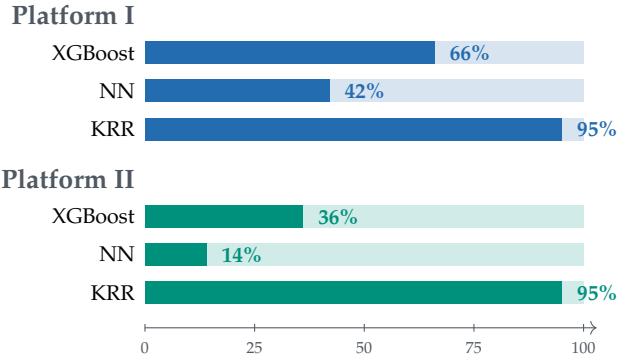
MSE reduction from target-only to TLCQM



Objective: Predict country-level MAU for a held-out app at Meta.

- Four source apps with ≈ 230 observations per app across two device platforms.

MSE reduction from target-only to TLCQM



Average test MSE (TLCQM standard errors in parentheses)

	TLCQM			TKRR	CDAR	DARC
	XGBoost	NN	KRR			
Platform I	**1.545 (0.198)**	**1.689 (0.146)**	**1.556 (0.207)**	2.204	3.183	36.380
Platform II	**1.364 (0.034)**	**1.841 (0.015)**	**1.377 (0.034)**	2.348	1.867	10.884

Main Takeaways of TLCQM

1. **Generate distributions, not point predictions,** from each source at target covariates.

2. **Match target quantiles** to correct discrepancies between $P^{(0)}(Y|X)$ and $P^{(k)}(Y|X)$, $k = 1, \ldots, K$.

3. **Train a standard predictor** on a target-aligned augmented dataset.

Main Takeaways of TLCQM

1. **Generate distributions, not point predictions,** from each source at target covariates.

2. **Match target quantiles** to correct discrepancies between $P^{(0)}(Y|X)$ and $P^{(k)}(Y|X)$, $k = 1, \ldots, K$.

3. **Train a standard predictor** on a target-aligned augmented dataset.

Thank you!

More details are in

https://arxiv.org/abs/2602.02358.

Bottom Line: Conditional quantile matching helps avoid negative transfer, while transfer bias governs downstream performance.

References

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Computation*, 14(1):21–41, 2002.

N. Sgouropoulos, Q. Yao, and C. Yastremiz. Matching a distribution by matching quantiles estimation. *Journal of the American Statistical Association*, 110(510):742–759, 2015.

X. Shen and N. Meinshausen. Engression: extrapolation through the lens of distributional regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(3):653–677, 2025.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

Y. Zhang, S. Wilkins-Reeves, W. Lee, and A. Hofleitner. Transfer learning through conditional quantile matching. *arXiv preprint arXiv:2602.02358*, 2026.

- *Two Source Domains:*

$$Y^{(1)} = \sin\left(3\theta^T X^{(1)}\right) + 1 + \epsilon, \quad Y^{(2)} = 2\cos\left(3\theta^T X^{(2)}\right) + 1 + \epsilon,$$

where $\theta = \left(1, \frac{1}{2}, \ldots, \frac{1}{6}\right)^T \in \mathbb{R}^6$ and $X^{(1)}, X^{(2)} \sim \mathcal{N}(\mathbf{1}_6, \boldsymbol{I}_6)$, $\epsilon \sim \mathcal{N}\left(0, \frac{1}{4}\right)$.

- *Target Domain:* $Y^{(0)} = \frac{1}{3}\sin\left(3\theta^T X^{(0)}\right) - 3 + \epsilon$ with $X^{(0)} \sim \mathcal{N}(\mathbf{0}_6, 0.25 \cdot \boldsymbol{I}_6)$ and $\epsilon \sim \mathcal{N}(0, 0.25)$.