

Transfer Learning Through Conditional Quantile Matching

Yikun Zhang¹

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

¹Department of Statistics, University of Washington

²Central Applied Science, Meta

New Horizons on Model Transportability and Data Integration

June 23, 2026

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 $k = 1, \dots, K$.
- 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 $k = 1, \dots, K$.

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 $k = 1, \dots, K$.
- 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 $k = 1, \dots, K$.

Covariate Shift (Shimodaira, 2000):

$$P^{(k)}(Y|X) = P^{(0)}(Y|X) \text{ but } P^{(k)}(X) \neq P^{(0)}(X).$$

Label Shift (Saerens et al., 2002):

$$P^{(k)}(X|Y) = P^{(0)}(X|Y) \text{ but } P^{(k)}(Y) \neq P^{(0)}(Y).$$

General Target Shift:

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

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

$\hat{P}^{(k)}(\cdot|X)$ for each
 $k = 1, \dots, K.$



- 1 Learn a generative model $\hat{P}^{(k)}(\cdot|x)$ 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

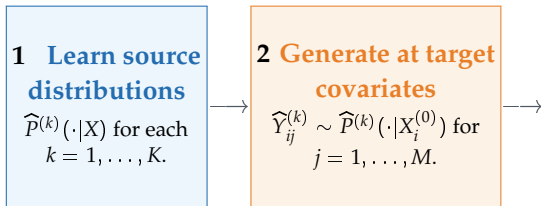
$\widehat{P}^{(k)}(\cdot|X)$ for each $k = 1, \dots, K$.

- 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 engrression (Shen and Meinshausen, 2025):

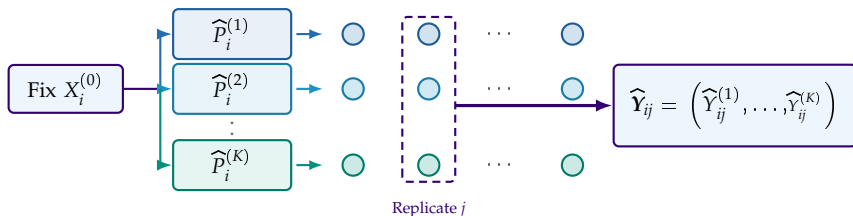
$$\widehat{g}^{(k)} \in \arg \min_{g \in \mathcal{G}} \frac{1}{n_k} \sum_{i=1}^{n_k} \left[\frac{1}{m} \sum_{j=1}^m \left| Y_i^{(k)} - g(X_i^{(k)}, \eta_{ij}) \right| - \frac{1}{2m(m-1)} \sum_{j=1}^m \sum_{j'=1}^m \left| g(X_i^{(k)}, \eta_{ij}) - g(X_i^{(k)}, \eta_{ij'}) \right| \right],$$

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

Proposed Transfer Learning Framework (TLCQM)

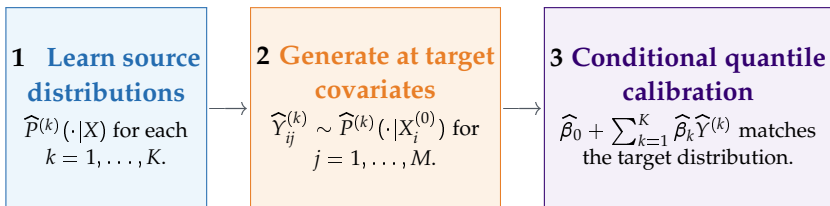


- 2 For each target covariate $X_i^{(0)}$, independently draw M synthetic response vectors:



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

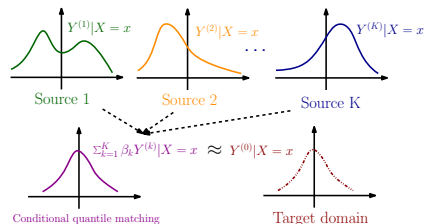
Proposed Transfer Learning Framework (TLCQM)



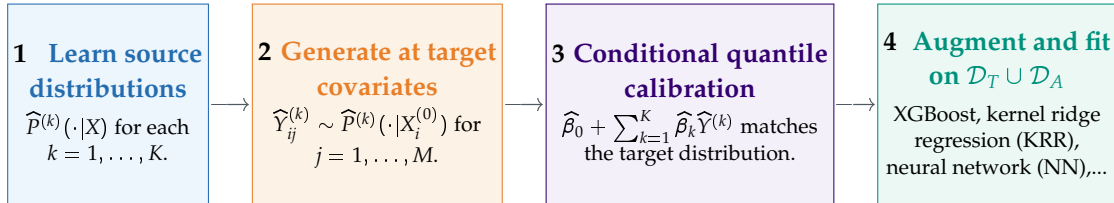
- 3 Compute the quantile-matching estimator (Sgouropoulos et al., 2015):

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

- $\widehat{V}_{ij} = \left(1, \widehat{Y}_{ij} \right)^T$, while $Y_{(1)}^{(0)} \leq \dots \leq Y_{(n_0)}^{(0)}$ and $\left(\beta^T \widehat{V} \right)_{(1)} \leq \dots \leq \left(\beta^T \widehat{V} \right)_{(n_0M)}$ are order statistics, respectively.



Proposed Transfer Learning Framework (TLCQM)



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

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

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 $k = 1, \dots, K$ to correct for covariate shift.

Theory: Excess Risk Decomposition

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

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

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

Theory: Excess Risk Decomposition

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

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

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

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

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

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

Theory: Quantile Matching Error

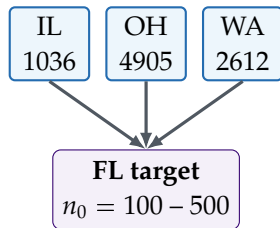
$$\begin{aligned}
 R(\widehat{f}^{(0,t)}) - R(f^{(0)}) &\lesssim \underbrace{\text{Rad}_N(\mathcal{F}) + \sqrt{\frac{K \log(1/\delta)}{N}}}_{\text{Standard generalization error}} + \underbrace{\frac{1}{N} \sum_{k=1}^K \|\widehat{w}_k - w_k\|_1}_{\text{Importance weight error}} + \underbrace{\sum_{k=1}^K \|\widehat{g}^{(k)} - g^{(k)}\|_\infty}_{\text{Distributional learning error}} \\
 &+ \underbrace{\inf_{\beta_* \in \mathcal{B}} \|\widehat{\beta} - \beta_*\|_1}_{\text{Quantile matching error}} + \underbrace{\inf_{\beta_* \in \mathcal{B}} \left(\int_0^1 [Q_{Y^{(0)}}(\alpha) - Q_{\beta_*^T V}(\alpha)]^2 d\alpha \right)^{1/2}}_{\text{Transfer bias}}.
 \end{aligned}$$

① $N = \sum_{k=0}^K n_k \gg n_0$ reduces the generalization/complexity error.

② $\inf_{\beta_* \in \mathcal{B}} \|\widehat{\beta} - \beta_*\|_1 = O\left(\sqrt{K \sum_{k=1}^K \|\widehat{g}^{(k)} - g^{(k)}\|_\infty}\right) + O_P\left(\sqrt{\frac{K \log \log n_0}{n_0}} + \sqrt{K} \left[\frac{\log \log n_0}{n_0} \cdot \inf_{\beta_* \in \mathcal{B}} \int_0^1 \{Q_{Y^{(0)}}(\alpha) - Q_{\beta_*^T V}(\alpha)\}^2 d\alpha\right]^{1/4}\right)$.

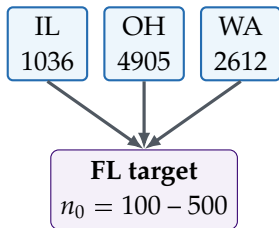
③ **Transfer bias** vanishes when $P^{(0)}(Y|X)$ lies in the convex hull of $\{P^{(k)}(Y|X)\}_{k=1}^K$.

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
1 rule 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.

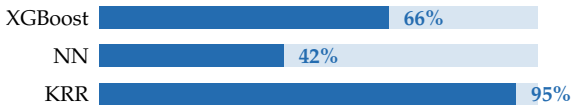


Objective: Predict country-level MAU for a held-out app at Meta.

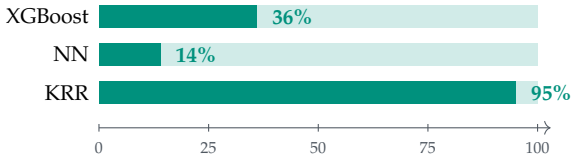
- Four source apps with ≈ 230 observations per app across two device platforms.

MSE reduction from target-only to TLCQM

Platform I



Platform II



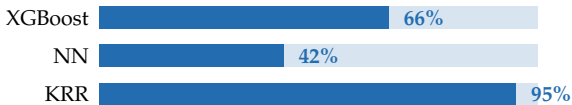


Objective: Predict country-level MAU for a held-out app at Meta.

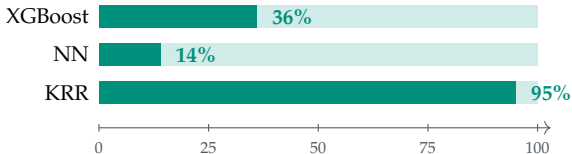
- Four source apps with ≈ 230 observations per app across two device platforms.

MSE reduction from target-only to TLCQM

Platform I



Platform II



Average test MSE (TLCQM standard errors in parentheses)

	XGBoost	TLCQM NN	KRR	TKRR	CDAR	DARC
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 $P^{(k)}(Y|X)$, $k = 1, \dots, K$.

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 $P^{(k)}(Y|X)$, $k = 1, \dots, K$.

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.

- 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.

Simulation: General Target Shift

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

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

