SYN-DIGITS: A Synthetic Control Framework for Calibrated Digital Twin Simulation

Grace Jiarui Fan [1] **Chengpiao Huang** [2] **Tianyi Peng** [3] **Kaizheng Wang** [2 4] **Yuhang Wu** [3]

Abstract

AI-based persona simulation—often referred to as *digital twin simulation*—is increasingly used for market research, recommender systems, and social sciences. Despite their flexibility, large language models (LLMs) often exhibit systematic bias and miscalibration relative to real human behavior, limiting their reliability. Inspired by *synthetic control* methods from causal inference, we propose **SYN-DIGITS** (**SYN**thetic Control Framework for Calibrated **DIGI**tal **T**win **S**imulation), a principled and lightweight calibration framework that learns latent structure from digital-twin responses and transfers it to align predictions with human ground truth. SYN-DIGITS operates as a post-processing layer on top of any LLM-based simulator and thus is model-agnostic. We develop a latent factor model with interpretable alignment conditions that identify the key factors governing calibration quality, and we systematically evaluate ten calibration methods across thirteen persona constructions, three LLMs, and two datasets. SYN-DIGITS supports both individual-level and distributional simulation for previously unseen questions and unobserved populations, with formal error decompositions that characterize when calibration succeeds. Experiments show that SYN-DIGITS achieves up to 50% relative improvements in individual-level correlation and 50–90% relative reductions in distributional discrepancy compared to uncalibrated baselines.

Author names are ordered alphabetically. [1]Finance Division, Columbia Business School [2]Department of IEOR, Columbia University [3]Decision, Risk, and Operations Division, Columbia Business School [4]Data Science Institute, Columbia University. Correspondence to: Yuhang Wu <yuhang.wu@columbia.edu>.

The 2nd Workshop on Connecting Low-rank Representations in AI at the 43rd International Conference on Machine Learning, Seoul, South Korea. Copyright 2026 by the author(s).

1. Introduction

Simulating human behavior is central to scientific inquiry and practical decision-making, yet it remains challenging due to the heterogeneity, context dependence, and latent structure of human responses. Classical approaches—agent-based models, cognitive architectures, discrete-choice models, and latent-variable frameworks—offer structured representations but face limitations in scalability and generality (McFadden, 1974; Laird et al., 1987; Epstein & Axtell, 1996; Bonabeau, 2002; Train, 2009; Lord, 2012).

Large language models (LLMs) have emerged as promising human behavior simulators—often called *digital twins* (DTs)—due to their ability to generate coherent, context-sensitive responses across diverse domains (Aher et al., 2023; Horton, 2023; Tranchero et al., 2024; Binz et al., 2025; Toubia et al., 2025). However, LLMs are usually not optimized for faithful reproduction of human response distributions, and naïve deployment typically produces systematic deviations including biases, overconfidence, and distributional concentration (Santurkar et al., 2023; Scherrer et al., 2023; Rossi et al., 2024; Gao et al., 2025; Li et al., 2025; Hullman et al., 2026).

Recent efforts to narrow this sim-to-real gap include fine-tuning on task-specific data and prompt engineering strategies such as persona descriptions and in-context examples (Cho et al., 2024; Binz et al., 2025; Cao et al., 2025; Kolluri et al., 2025; Peng et al., 2026). However, fine-tuning is computationally expensive and struggles in data-scarce settings common in practice, such as between cycles of online model updates, while prompt engineering lacks principled mechanisms for correcting systematic biases. Neither paradigm provides a unified, model-agnostic framework for correcting misalignment across tasks, questions, and populations. This motivates calibration methods that *complement* these approaches—methods that are lightweight, data-efficient, and amenable to theoretical analysis.

1.1. Overview

Motivated by these desiderata, we propose **SYN-DIGITS** (**SYN**thetic Control Framework for Calibrated **DIGI**tal **T**win **S**imulation), a post-hoc calibration layer for any LLM-

based simulator—whether zero-shot, prompt-engineered, or fine-tuned. To illustrate our approach, suppose we have n DTs $\tilde{p}_1, \ldots, \tilde{p}_n$ corresponding to n real individuals $p_1, \ldots, p_n$. Each DT $\tilde{p}_i$ is constructed by prompting an LLM with individual-specific attributes (e.g., demographics or historical behavior) of p_i. Furthermore, assume these DTs have been evaluated on m reference questions $q_1, \ldots, q_m$ (where human responses exist) and on a new question q_{m+1} (where human responses are unobserved). We denote the real and synthetic response matrices by $Y, \tilde{Y} \in \mathbb{R}^{n \times (m+1)}$, where the target column $Y_{m+1} \in \mathbb{R}^n$ that contains human responses to q_{m+1} is entirely unobserved, while the reference columns $Y_{1:m}$ and the full DT matrix $\tilde{Y}$ are observed. Our objective is to predict Y_{m+1}. Note that standard collaborative filtering or matrix completion on the human data Y alone is *not applicable* in this setting: since Y_{m+1} is entirely unobserved, there are no anchor entries in the target column for these methods to leverage. Existing approaches for predicting Y_{m+1} when it is *entirely unobserved* include:

1. **Naïve simulation:** directly prompt the digital twins to produce $\tilde{Y}_{m+1}$ and use it as a prediction for Y_{m+1}. This is the predominant approach in existing digital twin frameworks (Park et al., 2024; Toubia et al., 2025; Peng et al., 2026), but it inherits the systematic biases of LLM simulation.

2. **Fine-tuning:** use questions $q_1, \ldots, q_m$ to fine-tune the LLM before predicting q_{m+1}. This is computationally expensive and sometimes infeasible. Even when fine-tuning is affordable, it can be brittle in practice; for instance, Toubia et al. (2025) finds that fine-tuning can perform worse than naïve simulation.

3. **In-context learning:** provide $q_1, \ldots, q_m$ and associated responses as in-context examples when prompting the LLM. While cheaper than fine-tuning, this approach remains fragile and still inherits LLM simulation bias (see Section 2.3 and Table 1).

All three approaches above attempt to improve the raw LLM output but share a fundamental limitation: they do not exploit the cross-item structure between synthetic and real responses. Our key observation is that vertically *stacking* Y and $\tilde{Y}$ yields a $2n \times (m+1)$ matrix with a single *missing half-column* corresponding to the unobserved real responses Y_{m+1}. This viewpoint allows us to draw on the rich toolkit of matrix completion and synthetic control, which have a long history of success in recommender systems, healthcare, econometrics, and beyond (Abadie et al., 2010; Mazumder et al., 2010; Candes & Recht, 2012; Athey et al., 2021; Agarwal et al., 2025). Building on this idea, as illustrated in Figure 1, our contributions are:

1. **A principled calibration framework.** We introduce SYN-DIGITS, a lightweight, data-efficient, and model-agnostic post-hoc calibration layer for digital twin simulations. It can complement naïve simulation, fine-tuning, in-context learning, and other LLM-based approaches. Through paired latent factor models, we identify interpretable and empirically verifiable alignment conditions—row and column space inclusion between human and DT latent representations—that characterize when calibration succeeds.

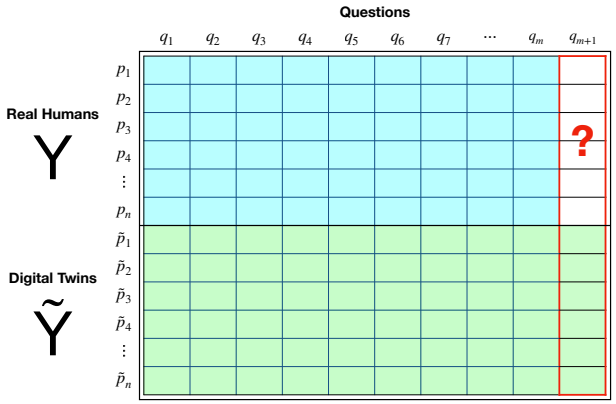


Figure 1. An illustration of our framework. Real responses Y are observed for $q_1, \ldots, q_m$, while digital-twin responses $\tilde{Y}$ are available for all questions including q_{m+1}. The goal is to recover the missing half-column of real responses to q_{m+1} in the stacked matrix, naturally motivating a matrix-completion or synthetic-control formulation.

2. **A comprehensive empirical study with strong performance.** We evaluate ten calibration methods on two datasets across thirteen persona constructions and three LLMs, providing practical guidance on method and persona selection. SYN-DIGITS consistently achieves strong gains on the task of predicting human responses to new questions, with the best method exhibiting up to 50% relative improvement over uncalibrated baselines.

3. **A unifying modeling perspective.** While Figure 1 illustrates the new-question setting, the same framework extends naturally to the new-user setting (Appendix D.2) as well as the task of distribution-level calibration (Section 5), where only aggregate distributional statistics are available rather than individual responses. Our latent factor model unifies all three settings and encompasses existing reweighting methods (Leng et al., 2024; Bui et al., 2025; Wang et al., 2026a) as special cases. Formal error bounds decompose prediction error into interpretable components that identify the key factors governing calibration quality.

1.2. Related Work

A growing literature deploys LLMs as human behavior simulators—or *digital twins*—for surveys, experiments, and agent-based simulations (Aher et al., 2023; Horton, 2023; Park et al., 2024; Binz et al., 2025; Toubia et al., 2025; Peng et al., 2026), while documenting systematic failure modes including distributional concentration, ideological biases, and prompt sensitivity (Santurkar et al., 2023; Scherrer et al., 2023; Rossi et al., 2024; Gao et al., 2025; Li et al., 2025; Hullman et al., 2026). To close the sim-to-real gap, existing approaches fall into three categories: (i) task-specific fine-tuning (Cho et al., 2024; Cao et al., 2025; Kolluri et al., 2025), which is computationally expensive and brittle in data-scarce settings; (ii) distributional reweighting of LLM-generated samples (Leng et al., 2024; Bui et al., 2025; Wang et al., 2026a), which calibrates aggregate statistics but not individual-level predictions; and (iii) inference procedures that account for distributional misalignment (Huang et al., 2025). None of these provides a unified, model-agnostic mechanism for individual-level calibration that generalizes across questions and populations.

Our framework draws on two classical toolkits. First, the synthetic control literature constructs weighted combinations of control units to approximate counterfactual outcomes (Abadie et al., 2010; Abadie, 2021). Second, matrix completion and latent factor methods recover missing entries in low-rank matrices (Mazumder et al., 2010; Candes & Recht, 2012; Athey et al., 2021). SYN-DIGITS repurposes these tools for the distinct goal of correcting systematic misalignment in digital-twin simulations. We note that *calibration* here refers to improving simulation fidelity, not confidence calibration in the sense of Platt scaling (Platt, 1999) or temperature scaling (Guo et al., 2017). A comprehensive related work discussion is deferred to Appendix A.

2. Framework and Motivation

2.1. Problem Setup

Consider tabular human response data, where rows correspond to individuals and columns correspond to questions or items. We have two response matrices $Y, \tilde{Y} \in \mathbb{R}^{n\times(m+1)}$—human and DT, respectively—indexed by n individuals and $m+1$ questions, and we denote the reference submatrices (the first m columns) by $Y_{1:m}, \tilde{Y}_{1:m} \in \mathbb{R}^{n\times m}$ (notation is summarized in Table 5 in the appendix). A crucial feature of this setting is that the target column $Y_{m+1} \in \mathbb{R}^n$ (the human responses to a new question q_{m+1}) is *entirely unobserved*. Our goal is to predict Y_{m+1}. This rules out standard collaborative filtering or matrix completion on Y alone, which require at least some observed entries in the target column to anchor the imputation. The DT responses $\tilde{Y}_{m+1}$ provide the only available signal about question q_{m+1}, making them indispensable for prediction.

Algorithm 1 Fit-and-transfer calibration

Require: $Y_{1:m} \in \mathbb{R}^{n\times m}$, $\tilde{Y} \in \mathbb{R}^{n\times(m+1)}$, model class $\mathcal{F}$
Ensure: Calibrated prediction $\hat{Y}_{m+1} \in \mathbb{R}^n$
1: Column-normalize $Y_{1:m}, \tilde{Y}$ (zero mean, unit variance) and impute missing entries via hard-thresholding SVD; store DT target moments $(\tilde{\mu}_{m+1}, \tilde{\sigma}_{m+1})$.
2: Fit $\hat{f} \in \operatorname{argmin}_{f\in\mathcal{F}} \|\tilde{Y}_{m+1} - f(\tilde{Y}_{1:m})\|_2^2$.
3: Transfer and de-normalize: $\hat{Y}_{m+1} = \hat{f}(Y_{1:m}) \cdot \tilde{\sigma}_{m+1} + \tilde{\mu}_{m+1}$.

2.2. Calibration Methods

The missing-half-column structure in Figure 1 admits two natural algorithmic paradigms. Because no single counterfactual learner uniformly dominates in panel settings (Athey et al., 2019), we evaluate ten calibration methods spanning both paradigms.

(a) Fit-and-transfer (Algorithm 1). Inspired by synthetic control (Abadie et al., 2010), this paradigm fits a predictive model f on the DT system—using $\tilde{Y}_{1:m}$ to predict $\tilde{Y}_{m+1}$—and transfers f to the human system. We consider six model classes: **Ridge** (ℓ_2-penalized regression), **Lasso** (ℓ_1-penalized, encouraging sparsity), **Elastic Net (EN)** (a convex combination of ℓ_1 and ℓ_2 penalties), **Neural Network (NN)** (a single-hidden-layer ReLU network allowing a nonlinear transfer mapping), **Synthetic Control (SC)** (Abadie et al., 2010) (weights constrained to the probability simplex), and **Synthetic Intervention (SI)** (Agarwal et al., 2025) (a linear mapping in the leading singular-value space of $\tilde{Y}$).

(b) Direct matrix completion (Algorithm 2). This paradigm vertically stacks Y and $\tilde{Y}$ and directly imputes the missing half-column. We consider four methods: **Hard SVD Impute (HSV)** and **Soft SVD Impute (SSV)**, which alternate between imputation and a rank-constrained or nuclear-norm-regularized SVD, respectively (Mazumder et al., 2010); **Alternating Least Squares (ALS)** (Hastie et al., 2015), which alternately solves for user and question factors; and **Synthetic Prior (SP)**, which uses $\tilde{Y}_{m+1}$ as a warm start for imputation on Y alone. In both paradigms, de-normalization uses the DT column moments as a proxy since the true human moments for q_{m+1} are unobserved (see Appendix F.5 for a bias analysis). Detailed descriptions and hyperparameters are in Appendix B.2.

2.3. Empirical Motivation on MovieLens

Dataset and persona. We evaluate all ten methods on the MovieLens-20M dataset (Harper & Konstan, 2015), which

Algorithm 2 Direct matrix completion calibration

Require: $Y_{1:m} \in \mathbb{R}^{n\times m}$, $\tilde{Y} \in \mathbb{R}^{n\times(m+1)}$, rank r
Ensure: Calibrated prediction $\hat{Y}_{m+1} \in \mathbb{R}^n$

1: Column-normalize and stack $Z = \begin{bmatrix} Y_{1:m} & ? \\ \tilde{Y}_{1:m} & \tilde{Y}_{m+1} \end{bmatrix}$, with ? the missing human target column; store $(\tilde{\mu}_{m+1}, \tilde{\sigma}_{m+1})$.
2: Impute the missing block of Z at rank r via matrix completion (iterative SVD, ALS, or nuclear-norm minimization).
3: Extract the top n entries of the imputed $(m+1)$-th column as $\hat{Y}^{\text{norm}}_{m+1}$ and de-normalize: $\hat{Y}_{m+1} = \hat{Y}^{\text{norm}}_{m+1} \cdot \tilde{\sigma}_{m+1} + \tilde{\mu}_{m+1}$.

contains ratings from 0.5 to 5 in half-point steps. We select the top 500 users and 500 movies by rating count, and split the 500 movies into 250 *persona movies* and 250 *prediction movies* (same split for all users). Each user's ratings on the persona movies serve as *persona information*, and their real and DT ratings on the prediction movies form the target matrices Y and $\tilde{Y}$ of size 500×250 with the same 22% missingness pattern. Columns of Y will be masked to mimic the missing-half-column structure in Figure 1 for evaluation.[1]

Persona simulation. We obtain DT ratings by querying GPT-4.1-mini at temperature 0 (Toubia et al., 2025). Each movie is described by its title, genre, and top 10 MovieLens tags; both the persona information and in-context examples are provided as lists of title–genre–tags–rating tuples (full prompt template in Appendix C.1).

Baselines and evaluation. We compare against two LLM baselines that differ only in the field `{in_context_ratings}`. For a fixed target `{movie}`, the **zero-shot (ZS)** baseline leaves `{in_context_ratings}` empty—so the LLM rates each target movie from the persona ratings alone—while **in-context (IC)** baseline fills it with the user's ground-truth ratings on the other 249 prediction movies. We evaluate all methods under a *leave-one-question-out* protocol: among the 250 prediction movies, each column is held out in turn and predicted from the remaining 249, mirroring the deployment setting of Figure 1 in which the held-out column plays the role of q_{m+1} and the remaining 249 play the role of $q_1, \ldots, q_m$. In the notation of Section 2.1, each fold has $n = 500$ and $m = 249$. Performance is measured by the average Pearson correlation between predicted and true responses.

Importantly, IC and our calibration methods have access to the same underlying information—persona ratings plus the 249 held-in prediction ratings—but use it differently: IC feeds the 249 ratings into the prompt as context, while the calibration methods use them as the reference matrix $Y_{1:m}$ in Algorithms 1 and 2 to learn a DT-to-human mapping that is then applied to the held-out column's DT prediction.

Calibration consistently outperforms baselines. Table 1 shows that all ten calibration methods substantially outperform both the zero-shot baseline (0.349) and the in-context baseline (0.406). Even the weakest calibration method (SP, 0.454) exceeds IC by 12%, while the best (EN, 0.524) improves over ZS by 50% and over IC by 29%. Fit-and-transfer methods generally outperform direct matrix completion, and linear methods (Ridge, EN, SI) outperform the nonlinear NN, suggesting that the transfer mapping is approximately linear.

Calibration vs. richer context. IC and calibration access the same ground-truth ratings but use them differently—IC as prompt context, calibration to learn a structural mapping. The large gap (IC: +16% vs. best calibration: +50%) suggests the sim-to-real discrepancy is a *structural misalignment* rather than an information deficiency.

DT structure transfers despite individual bias. The key intuition is that even though individual DT predictions are biased, the *inter-question structure* can be approximately preserved between humans and DTs, and a mapping learned on DT data can therefore generalize to real human data. We now formalize this through a latent factor model.

3. A Latent Factor Framework

For brevity, throughout Section 3 we let Y and $\tilde{Y}$ denote the $n \times m$ reference submatrices $Y_{1:m}$ and $\tilde{Y}_{1:m}$ from Section 2.1; the target columns are written separately as $Y_v := Y_{m+1}$ and $\tilde{Y}_v := \tilde{Y}_{m+1}$.

3.1. A Latent Factor Model

We assume that there exist latent user embeddings $u_i \in \mathbb{R}^d$, $i \in [n]$, and latent question embeddings $v_j \in \mathbb{R}^d$, $j \in [m]$, such that user i's response to question j is represented as:

$$Y_{ij} = \langle u_i, v_j \rangle + \varepsilon_{ij}, \tag{1}$$

where ε_{ij} is random noise. Let $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{m\times d}$ denote matrices whose rows are $\{u_i\}_{i=1}^n$ and $\{v_j\}_{j=1}^m$, respectively, and let $\mathcal{E} = (\varepsilon_{ij}) \in \mathbb{R}^{n\times m}$. Then (1) can be written compactly as:

$$Y = UV^\top + \mathcal{E}. \tag{2}$$

This formulation is standard in recommender systems, matrix completion, and panel causal inference (Koren et al.,

[1] Code and data available at https://github.com/yw3453/syn-digits.

Table 1. New-question prediction on MovieLens: average Pearson correlation, standard errors, and percentage improvements (%Δ) over the zero-shot (ZS) and in-context (IC) baselines. The best method (EN) improves over ZS by 50% and over IC by 29%.

	Baselines		**Fit-and-transfer**						**Matrix Completion**			
	ZS	IC	Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
Corr.	.349	.406	.511	.497	**.524**	.504	.457	.515	.473	.511	.512	.454
S.E.	.005	.005	.005	.006	**.005**	.005	.005	.005	.008	.005	.005	.005
%Δ ZS	—	+16	+46	+42	**+50**	+44	+31	+48	+36	+46	+47	+30
%Δ IC	—	—	+26	+22	**+29**	+24	+13	+27	+17	+26	+26	+12

2009; Candes & Recht, 2012; Xiong & Pelger, 2023). Analogously, we model the DTs' responses as $\tilde{Y} = \tilde{U}\tilde{V}^\top + \tilde{\mathcal{E}}$, where $\tilde{U}$ and $\tilde{V}$ are latent embeddings induced by the LLM and $\tilde{\mathcal{E}}$ is noise. SVD diagnostics confirm that both Y and $\tilde{Y}$ are approximately low-rank on MovieLens and Twin-2K-500, supporting the existence of low-dimensional latent structure (Figure 4 in the appendix).

Row space inclusion condition. A key question is: when does a mapping learned on DT data transfer to human data? In the noiseless setting, $Y = UV^\top$ and $\tilde{Y} = \tilde{U}\tilde{V}^\top$ with $\mathrm{rank}(Y) = \mathrm{rank}(\tilde{Y}) = d$. Let $v, \tilde{v} \in \mathbb{R}^d$ denote the latent embeddings of the new question in the human model and the DT model, respectively, and denote the unobserved human responses by $Y_v = Uv$ and the observed DT responses $\tilde{Y}_v = \tilde{U}\tilde{v}$. Then, given $\tilde{Y}\beta = \tilde{Y}_v$, exact transfer $Y\beta = Y_v$ holds whenever a *row space inclusion condition* holds:

$$\mathrm{Row}([V^\top, v]) \subseteq \mathrm{Row}([\tilde{V}^\top, \tilde{v}]). \tag{3}$$

This requires the LLM's latent question geometry to span at least the human question geometry, while allowing imperfect fidelity ($\tilde{V} \neq V$). Intuitively, if the DT question embeddings are sufficiently rich to represent any human question, then a regression fit on the DT system transfers exactly to the human system—a direct analogue of the identifying condition in classical synthetic control (Abadie et al., 2010; Agarwal et al., 2025) and related transfer-learning methods for panel factor models (Duan et al., 2024).

3.2. Empirical Evidence for Row Space Alignment

Though (3) is unlikely to hold exactly in practice, the strong empirical performance on MovieLens (Table 1) suggests that it holds approximately. We assess this by measuring the similarity between the row spaces of $V^\top$ and $\tilde{V}^\top$ via the cosine of principal angles and the projection-Frobenius norm on the leading $r_{\max} = r+2$ singular directions, where r is the effective rank estimated using a rank-constrained iterative SVD scheme with validation. As baselines, we include (i) a random $\mathcal{N}(0,1)$ matrix and (ii) a column-wise shuffled version of the real matrix, which preserves marginal column distributions. Figure 2 shows that the DT and human row spaces align substantially more closely than random baselines, providing empirical support for the approximate validity of (3).

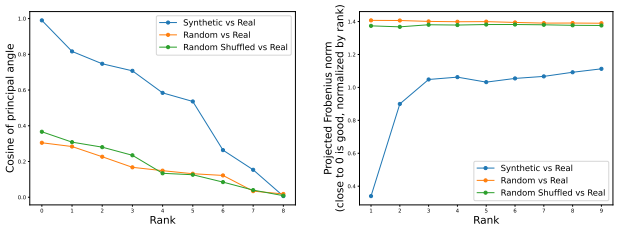


Figure 2. Row space alignment on MovieLens (zero-shot DT baseline). Cosine of principal angles (left) and projection-Frobenius norm (right) between the row spaces of $V^\top$ and $\tilde{V}^\top$, restricted to the leading $r_{\max} = r+2$ singular directions. The DT row space is substantially more aligned with the human row space than random baselines. Higher cosine similarity and lower Frobenius distance indicate closer alignment.

Interpreting the alignment structure. Figure 2 reveals a graded alignment pattern that is informative about the LLM's internal representations. The leading singular directions, which capture the dominant axes of variation in user preferences, show near-perfect cosine similarity between the human and DT row spaces. This indicates that the LLM captures well the primary structure of how questions relate to each other (e.g., genre preferences, quality signals). As we move to later singular directions, alignment degrades, reflecting finer-grained patterns of human taste that the LLM does not fully reproduce. Crucially, the projection-Frobenius norm remains far below the random baselines even at the highest ranks considered, suggesting that the LLM's latent question space is a noisy but meaningful superset of the human one.

Approximate alignment suffices. The row space inclusion (3) need not hold exactly for calibration to succeed. We now formalize this in the noisy setting. In the noisy model, the unobserved human responses and the observed DT responses to the new question are $Y_v = Uv + \mathcal{E}_v \in \mathbb{R}^n$ and $\tilde{Y}_v = \tilde{U}\tilde{v} + \tilde{\mathcal{E}}_v \in \mathbb{R}^n$, where $\mathcal{E}_v$ and $\tilde{\mathcal{E}}_v$ are noise vectors.

Algorithm 1 with ridge regression fits

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^m} \|\tilde{Y}_v - \tilde{Y}\beta\|_2^2 + \lambda\|\beta\|_2^2, \quad (4)$$

on the DT system, then predicts $\hat{Y}_v = Y\hat{\beta}$ on the human system. To analyze the error, assume the new question embedding can be approximately represented by existing question embeddings: $v = Vb + e$ for some $b \in \mathbb{R}^m$ and residual $e \in \mathbb{R}^d$. Define

$$\tilde{\Sigma} = \tfrac{1}{n}\mathbb{E}[\tilde{Y}^\top \tilde{Y}], \quad \tilde{\gamma} = \tfrac{1}{n}\mathbb{E}[\tilde{Y}^\top \tilde{Y}_v],$$
$$\hat{\tilde{\Sigma}} = \tfrac{1}{n}\tilde{Y}^\top \tilde{Y}, \quad \hat{\tilde{\gamma}} = \tfrac{1}{n}\tilde{Y}^\top \tilde{Y}_v,$$
$$\beta^* = (\tilde{\Sigma} + \lambda I)^{-1}\tilde{\gamma}, \quad r = Y_v - Y\beta^*,$$

where the expectation is over the noise $\mathcal{E}$, $\tilde{\mathcal{E}}$, $\mathcal{E}_v$, and $\tilde{\mathcal{E}}_v$. The following error bound holds; the proof is in Appendix G.1.

Theorem 3.1 (Error on new question)**.**

$$\|\hat{Y}_v - Y_v\|_2 \le \|r\|_2 + \frac{\|Y\|_2\big(\|\hat{\tilde{\Sigma}} - \tilde{\Sigma}\|_2\|\beta^*\|_2 + \|\hat{\tilde{\gamma}} - \tilde{\gamma}\|_2\big)}{\sigma_{\min}(\hat{\tilde{\Sigma}}) + \lambda},$$

with $\|r\|_2 \le \|Ue\|_2 + \|\mathcal{E}_v - \mathcal{E}b\|_2 + \frac{\|Y\|_2}{\sigma_{\min}(\tilde{\Sigma})+\lambda}\big(\lambda\|b\|_2 + \|\tilde{\gamma} - \tilde{\Sigma}b\|_2\big)$.

Error decomposition. The bound decomposes into a *structural error* $\|r\|_2$ (misalignment between the new question's latent embedding and the span of existing questions, plus noise) and an *estimation error* (finite-sample error in $\tilde{\Sigma}$, $\tilde{\gamma}$, vanishing as $n \to \infty$). The structural term is small when the DT inter-question covariance aligns with the human one (cf. (3)), i.e., *full digital-twin fidelity is not required for meaningful transfer.*

4. Systematic Evaluation on Twin-2K-500

Having established that calibration works across a broad family of methods on MovieLens and developed a theoretical framework to explain why, we now evaluate systematically on a second dataset with thirteen distinct persona constructions, examining how the choice of LLM, prompt format, and prompting strategy interact with calibration.

4.1. Dataset and Experimental Setup

Dataset. The Twin-2K-500 dataset (Toubia et al., 2025) contains 2,058 U.S. participants and their responses to 123 demographic, psychological, behavioral, and economic questions, with 23% missing values. The original paper provides zero-shot digital twin simulations across thirteen persona constructions, making it an ideal testbed for evaluating calibration across diverse simulation strategies. Columns of Y will be masked to mimic the missing-half-column structure in Figure 1 for evaluation.

Persona constructions. The thirteen constructions vary along four axes—LLM backbone (GPT-4.1-mini, GPT-4.1, Gemini-Flash-2.5), persona format (free-text, JSON, summary, demographics-only), prompting strategy (zero-shot, chain-of-thought, repeat, predicted output), and persona content (full survey, summary, demographics). All use temperature 0 except one variant at 0.7; one construction uses a fine-tuned GPT-4.1-mini. Exact details are in Appendix C.2 and Toubia et al. (2025).

Diagnostics. Using the default persona construction (Text, GPT-4.1-mini), SVD diagnostics confirm that Twin-2K-500 exhibits the same low-rank structure as MovieLens (Figure 4), and the row space alignment between human and DT responses closely mirrors the MovieLens pattern (Figure 5), providing further empirical support for condition (3). Both diagnostic figures are in Appendix C.3.

Leave-one-question-out evaluation. We follow the same protocol as Section 2.3: each of 123 questions is held out in turn (n = 2,058, m = 122). The baseline (BL) is the raw DT prediction; calibration methods use ground-truth human responses on the 122 reference questions to correct $\tilde{Y}_{m+1}$. We omit the IC baseline since it is consistently outperformed by all calibration methods on MovieLens (Table 1). Performance is measured by average Pearson correlation. Results using Fisher's z-transformation (Silver & Dunlap, 1987; Park et al., 2024) are qualitatively identical.

4.2. Results

Table 2 reports the results. The central empirical finding is that **calibration consistently and substantially improves fidelity**: the best method outperforms the uncalibrated baseline for all thirteen constructions, with relative improvements ranging from 8.5% (JSON + PO, GPT-4.1) to over 400% (fine-tuned GPT-4.1-mini, from .048 to .243). This confirms the MovieLens findings (Section 2.3) on a more diverse testbed. We now examine the patterns in more detail.

Stability of method rankings across personas. The relative ranking of calibration methods is remarkably stable across the thirteen persona constructions, even as baseline quality varies dramatically (from .048 to .205). EN is the best method in 8 of 13 constructions, while Ridge, SSV, and ALS are also consistently effective. Synthetic control and synthetic prior consistently underperform, indicating that the simplex constraint and DT-only warm starts are insufficient without exploiting inter-question transfer structure. This stability simplifies practical method selection: practitioners can choose EN as a robust default.

Calibration as an equalizer. The %Δ column reveals a striking inverse relationship between baseline quality and

Table 2. New-question prediction on Twin-2K-500: average Pearson correlation for thirteen persona constructions × ten calibration methods. Mean above, standard error in parentheses below. BL = uncalibrated DT baseline. %Δ = relative improvement of the best method (highlighted) over BL.

		Fit-and-transfer						Matrix Completion				
Persona	**BL**	**Ridge**	**Lasso**	**EN**	**NN**	**SC**	**SI**	**HSV**	**SSV**	**ALS**	**SP**	**%Δ**
Text, GPT-4.1-mini	.168 (.015)	.200 (.021)	.195 (.021)	**.204** (.021)	.198 (.020)	.147 (.013)	.190 (.017)	.170 (.017)	.202 (.018)	.201 (.018)	.121 (.016)	+21.4
Text, Gemini-Flash-2.5	.188 (.016)	.237 (.020)	.234 (.020)	**.242** (.020)	.215 (.020)	.158 (.013)	.200 (.019)	.179 (.018)	.221 (.019)	.220 (.019)	.127 (.015)	+28.7
JSON, GPT-4.1-mini	.175 (.016)	.211 (.021)	.195 (.021)	**.214** (.021)	.207 (.020)	.158 (.013)	.210 (.018)	.185 (.018)	.213 (.018)	.213 (.018)	.137 (.015)	+22.3
JSON, GPT-4.1	.205 (.018)	.213 (.019)	.201 (.019)	.220 (.019)	.215 (.019)	.156 (.013)	.211 (.017)	.181 (.017)	**.223** (.018)	.222 (.018)	.124 (.016)	+8.78
Text + CoT, GPT-4.1-mini	.165 (.015)	.227 (.022)	.221 (.022)	**.238** (.022)	.228 (.021)	.159 (.013)	.213 (.019)	.182 (.019)	.224 (.019)	.223 (.019)	.121 (.016)	+44.2
Text + repeat, GPT-4.1-mini	.173 (.015)	.226 (.022)	.220 (.022)	**.235** (.022)	.229 (.021)	.155 (.013)	.212 (.018)	.193 (.018)	.221 (.019)	.220 (.019)	.131 (.015)	+35.8
Text, temp=0.7, GPT-4.1-mini	.158 (.015)	.199 (.021)	.195 (.021)	**.204** (.021)	.198 (.020)	.144 (.013)	.197 (.017)	.165 (.018)	.204 (.018)	.204 (.018)	.113 (.016)	+29.1
JSON + PO, GPT-4.1-mini	.158 (.014)	.203 (.020)	.191 (.019)	**.211** (.020)	.201 (.019)	.153 (.013)	.188 (.017)	.151 (.017)	.207 (.019)	.208 (.019)	.108 (.015)	+33.5
JSON + PO, GPT-4.1	.200 (.017)	.208 (.019)	.200 (.019)	**.217** (.019)	.212 (.019)	.151 (.012)	.202 (.019)	.150 (.018)	.215 (.018)	.214 (.018)	.143 (.015)	+8.50
Fine-tuned GPT-4.1-mini	.048 (.008)	**.243** (.019)	.220 (.018)	.233 (.018)	.232 (.018)	.150 (.012)	.231 (.018)	.201 (.018)	.226 (.017)	.226 (.017)	.101 (.016)	+406
Demographics only, GPT-4.1-mini	.122 (.012)	.196 (.021)	.184 (.021)	.204 (.021)	.201 (.020)	.157 (.013)	.193 (.019)	.155 (.016)	.212 (.018)	**.213** (.018)	.103 (.015)	+74.6
Summary, GPT-4.1-mini	.148 (.013)	.222 (.020)	.211 (.020)	.227 (.020)	.221 (.020)	.169 (.013)	.215 (.018)	.174 (.015)	.232 (.018)	**.233** (.018)	.111 (.016)	+57.4
Summary + JSON, GPT-4.1-mini	.102 (.011)	.223 (.019)	.212 (.019)	.229 (.019)	.224 (.019)	.164 (.013)	.214 (.017)	.184 (.017)	**.235** (.018)	.232 (.018)	.130 (.015)	+130

relative calibration gain. Constructions with the weakest baselines (fine-tuned GPT-4.1-mini, .048; summary + JSON, .102) exhibit the largest improvements (+406%, +130%), while those with the strongest baselines (JSON GPT-4.1, .205) show smaller gains (+8.78%). More importantly, post-calibration correlations are tightly clustered (.204–.243) compared to baselines (.048–.205), suggesting calibration acts as an *equalizer*. The most striking example is fine-tuned GPT-4.1-mini: it has the *worst* baseline yet achieves the *best* post-calibration correlation, indicating that fine-tuning produces structured response patterns that calibration can exploit even when direct generalization fails.

Ablation studies and additional details. We complement the main results with several robustness checks: method rankings are consistent across supplementary metrics (RMSE, MAE, Spearman's ρ; Appendix F.1); null-DT diagnostics confirm that gains are driven by genuine DT signal (Appendix F.2); results are stable across imputation strategies (Appendix F.3); and all methods run in seconds per fold (Appendix F.4).

Adaptive transfer. Algorithm 1 always transfers the fitted model, but if the model fits the DT system poorly for a given question, transfer may be counterproductive. A simple safeguard uses the synthetic training MSE as a fit diagnostic: if it exceeds a threshold τ, we revert to the uncalibrated DT prediction. Oracle analysis shows that adaptive transfer can achieve up to 50% improvement over the baseline on Twin-2K-500, compared to 19–21% for always-transfer calibration. Developing principled threshold selection (e.g., cross-validated synthetic error) is an important direction for future work. Full details and figures are in Appendix D.1.

Extension to new-user prediction. By symmetry of the latent factor model, alignment in the *user* embedding space enables predicting responses for previously unseen users. Given a new user's DT responses $\tilde{Y}^{(u)} \in \mathbb{R}^m$, we apply Algorithm 1 to the *transposed* response matrices, swapping the roles of users and questions. The sufficient condition for exact transfer then becomes a *column space inclusion condition*:

$$\mathrm{Col}(U) \subseteq \mathrm{Col}(\tilde{U}), \tag{5}$$

Table 3. New-user prediction: average Pearson correlation, standard errors, and percentage improvements (%Δ) over the zero-shot (ZS) baseline. ML = MovieLens, 2K = Twin-2K-500, BL = baseline. The best method (NN) improves over BL by 53% on ML and 3.5% on 2K.

		BL	**Fit-and-transfer**						**Matrix Completion**			
		ZS	Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
ML	Corr.	.246	.373	.372	.373	**.377**	.375	.361	.372	.372	.372	.366
	S.E.	.006	.006	.006	.006	**.006**	.006	.006	.006	.006	.006	.007
	%Δ	—	+52	+51	+52	**+53**	+52	+47	+51	+51	+51	+49
2K	Corr.	.847	.875	.873	.875	**.877**	.874	.875	.873	.873	.873	.863
	S.E.	.002	.002	.002	.002	**.002**	.002	.002	.002	.002	.002	.002
	%Δ	—	+3.3	+3.1	+3.3	**+3.5**	+3.2	+3.3	+3.1	+3.1	+3.1	+1.9

requiring the LLM's latent user geometry to span the human one; empirical diagnostics confirm this holds approximately (Appendix D.2). Table 3 reports leave-one-user-out results: all ten methods improve over the baseline on both datasets, with the best method (NN) achieving a 53% relative improvement on MovieLens (from .246 to .377) and 3.5% on Twin-2K-500 (from .847 to .877). The smaller Twin-2K-500 gain reflects an already-strong baseline—the rich persona, built from a user's full survey, already captures most individual-level variation, leaving little headroom—whereas predicting responses to a *new question* requires extrapolating inter-question structure, where calibration adds the most value. Full diagnostics and discussion are in Appendix D.2.

5. Distribution-level Calibration

In many applications like market research or opinion polling, individual responses are unavailable or unnecessary, and the primary object of interest is the *marginal distribution* of responses across a population. We now show that our latent factor framework also provides a principled foundation for this distributional calibration task, and offers a theoretical understanding of existing methods (Leng et al., 2024; Bui et al., 2025; Wang et al., 2026a). We also provide a comprehensive empirical evaluation on MovieLens and OpinionQA, demonstrating that SYN-DIGITS achieves up to 90% relative reductions in distributional discrepancy compared to uncalibrated baselines.

5.1. The Weighted Ensemble Method

Suppose we observe only the marginal distribution P_j for each training question $j \in [m]$ and wish to predict P_{m+1} using $\{P_j\}_{j=1}^m$ and DT responses $\tilde{Y} \in \mathbb{R}^{n\times(m+1)}$, where responses take values in $[K]$. We represent the human population as a *weighted ensemble* of n DTs plus K dummy twins (the k-th always selects answer k) to ensure full support. The ensemble distribution for question j is

$$\hat{P}_j(w,\pi) = \sum_{i=1}^{n} w_i \delta_{\tilde{Y}_{ij}} + \sum_{k=1}^{K} \pi_k \delta_k, \tag{6}$$

where (w,π) is constrained to the probability simplex Δ^{n+K-1}. We calibrate (w,π) by matching (6) to P_j across training questions:

$$(\hat{w},\hat{\pi}) = \operatorname{argmin}_{(w,\pi)\in\Delta^{n+K-1}} \frac{1}{m}\sum_{j=1}^{m} D\big(P_j \,\|\, \hat{P}_j(w,\pi)\big), \tag{7}$$

where D is a distribution discrepancy measure (e.g., total variation (TV) or KL divergence). The predicted distribution for the new question is then

$$\hat{P}_{m+1} = \sum_{i=1}^{n} \hat{w}_i \delta_{\tilde{Y}_{i,m+1}} + \sum_{k=1}^{K} \hat{\pi}_k \delta_k. \tag{8}$$

The procedure is summarized in Algorithm 3 (in Appendix E.2).

5.2. Theoretical Analysis

We now develop a formal analysis of Algorithm 3 to provide a theoretical foundation for understanding existing reweighting methods (Leng et al., 2024; Bui et al., 2025; Wang et al., 2026a) and to identify the sources of distributional prediction error that determine when distributional calibration succeeds. We work under three assumptions (formal statements in Appendix E.1): (i) a linear probability model where response probabilities factorize as inner products of user and *shared* question embeddings, making it possible for reweighting DTs to approximate the human population; (ii) i.i.d. user embeddings with the DT distribution supported on a finite set; and (iii) a span condition requiring the new question's embedding to lie in the convex hull of training question embeddings. Under these assumptions, the following error bound holds (proof in Appendix G.2). We denote by $\tilde{P}_j(\cdot \mid \tilde{u}_i)$ the conditional distribution of $\tilde{Y}_{ij}$ given $\tilde{u}_i$.

Theorem 5.1 (Distributional prediction error bound)**.** *Under Assumptions E.1–E.3, let* $(\hat{w}, \hat{\pi})$ *solve* (7) *with* $D = \mathrm{TV}$. *For* $\alpha \in (0,1)$, $A > 0$, *with probability at least* $1-\alpha$,

$$\mathrm{TV}\Big(P_{m+1}, \sum_{i=1}^{n} \hat{w}_i \tilde{P}_{m+1}(\cdot \mid \tilde{u}_i) + \sum_{k=1}^{K} \hat{\pi}_k \delta_k\Big)$$
$$\leq m\|c\|_\infty \left[\inf_{\nu} \frac{1}{m}\sum_{j=1}^{m} \mathrm{TV}(j, v, u) + \sqrt{\frac{3A^2(K+\log(4/\alpha))}{n}}\right],$$

where $\mathrm{TV}(j,v,u) = \mathrm{TV}(P_j, \sum_u \nu(u)\tilde{P}_j(\cdot \mid u))$ *and the infimum is over* $\nu : \mathcal{U} \to [0,1]$ *with* $\sum_u \nu(u) = 1$, $\nu(u) \leq A\tilde{\mu}(u)$.

Theorem 5.1 decomposes the prediction error into two interpretable components. The first term (the infimum) captures the *fundamental sim-to-real gap*: the best achievable distributional match between the human response distribution and a population-level reweighting of digital twins. This term is zero when the DT population can perfectly represent the human population through reweighting, and is positive otherwise, reflecting an irreducible misalignment. The second term, $O(\sqrt{K/n})$, is the statistical error from finite sampling, which matches the minimax-optimal rate for estimating a K-category distribution from n samples under total variation.

The multiplicative factor $m\|c\|_\infty$ captures how well the new question is represented by the training questions. When $c_j = 1/m$ for all j (the new question is a uniform mixture of training questions), $m\|c\|_\infty = 1$ and there is no degradation. In the worst case where the new question depends on a single training question, $m\|c\|_\infty = m$, reflecting a loss from extrapolation. This mirrors the role of the row space condition in the individual-level analysis (Section 3.2): both require the new question to be well-represented within the span of existing questions.

5.3. Empirical Evaluation

Setup. We evaluate on two datasets using the 2,058 digital twins from Twin-2K-500 (Summary, GPT-4.1-mini persona): **MovieLens** (500 movies, $K = 10$ rating levels) and **OpinionQA** (Santurkar et al., 2023) (489 Likert-scale survey questions, $K = 5$). We measure performance using seven discrepancy measures (TV, χ^2, KL, Hellinger, KS, CDF ℓ_1, CDF ℓ_2; definitions in Appendix E.4). Questions are split 80/20 into training and test sets, and weights (w, π) are optimized via mirror descent. The baseline is the uniform-weight ensemble ($w_i = 1/n$, $\pi_k = 0$). Full details on DT generation, ensemble variants, and prompt templates are in Appendix E.3.

Results. Table 4 summarizes the distributional prediction results (trained with CDF ℓ_1 loss, personas + dummy twins) for both MovieLens and OpinionQA; full cross-metric results across all training objectives and ensemble variants are in Tables 8 and 9 in the appendix. In all cases, the calibrated ensemble substantially outperforms the uniform-weight baseline, achieving 50% to 90% relative reductions in distributional divergence. Figure 3 illustrates this on a held-out MovieLens question: the calibrated ensemble closely tracks the true rating distribution, while the uniform-weight baseline exhibits systematic bias and missing support. Additional analysis on sensitivity to training objective, the role of dummy twins, variance retention, and scalability in n can be found in Appendix E.

Table 4. Distributional prediction results (CDF ℓ_1 loss, personas + dummy twins). ML = MovieLens, OQA = OpinionQA, Cal. = calibrated ensemble, BL = uniform-weight baseline. Lower is better. Full results in Tables 8 and 9.

		TV	χ^2	KL	Hell.	KS	ℓ_1	ℓ_2
ML	Cal.	.188	23.5	.212	.036	.123	.415	.041
	BL	.381	28600	2.44	.176	.280	.922	.189
OQA	Cal.	.178	.298	.123	.031	.154	.271	.045
	BL	.350	11800	1.30	.137	.303	.520	.154

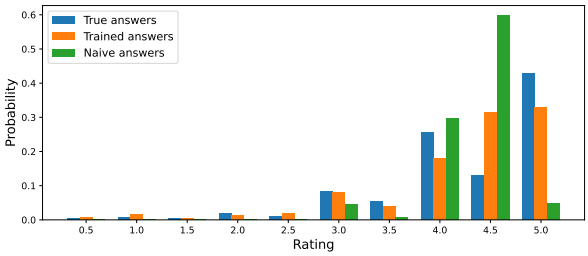


Figure 3. Distributional calibration on a held-out MovieLens question (trained with CDF ℓ_1 loss). True rating distribution (blue), calibrated ensemble (orange), and uniform-weight baseline (green). The calibrated ensemble closely matches the true distribution, whereas the baseline shows systematic bias and missing support.

6. Discussion

We present SYN-DIGITS, a practical, model-agnostic post-hoc calibration framework that consistently improves digital twin simulations—up to 50% relative gains in individual-level correlation and 50–90% reductions in distributional discrepancy—across ten methods, thirteen persona constructions, three LLMs, and two datasets. Simple linear methods (Ridge, elastic net) are strong defaults; calibration acts as an equalizer across persona constructions. Current limitations include restriction to structured numerical responses and dependence on reference questions spanning a sufficiently rich latent space. Future directions include free-text responses, semantic side information, principled adaptive transfer, and tighter theoretical bounds.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abadie, A. Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2):391–425, 2021.

Abadie, A. and Gardeazabal, J. The economic costs of conflict: A case study of the basque country. *American economic review*, 93(1):113–132, 2003.

Abadie, A. and Zhao, J. Synthetic controls for experimental design. *arXiv preprint arXiv:2108.02196*, 2021.

Abadie, A., Diamond, A., and Hainmueller, J. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American Statistical Association*, 105(490):493–505, 2010.

Agarwal, A., Dahleh, M., Shah, D., and Shen, D. Causal matrix completion. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 3821–3826. PMLR, 2023.

Agarwal, A., Shah, D., and Shen, D. Synthetic interventions: Extending synthetic controls to multiple treatments. *Operations Research*, 2025.

Aher, G., Arriaga, R. I., and Kalai, A. T. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Athey, S., Bayati, M., Imbens, G., and Qu, Z. Ensemble methods for causal effects in panel data settings. In *AEA Papers and Proceedings*, volume 109, pp. 65–70. American Economic Association, 2019.

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., and Khosravi, K. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116(536):1716–1730, 2021.

Ben-Michael, E., Feller, A., and Rothstein, J. The augmented synthetic control method. *Journal of the American Statistical Association*, 116(536):1789–1803, 2021.

Binz, M., Akata, E., Bethge, M., Brändle, F., Callaway, F., Coda-Forno, J., Dayan, P., Demircan, C., Eckstein, M. K., Éltető, N., et al. A foundation model to predict and capture human cognition. *Nature*, pp. 1–8, 2025.

Bonabeau, E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3):7280–7287, 2002.

Bui, N., Nguyen, H. T., Kumar, S., Theodore, J., Qiu, W., Nguyen, V. A., and Ying, R. Mixture-of-personas language models for population simulation. *arXiv preprint arXiv:2504.05019*, 2025.

Candes, E. and Recht, B. Exact matrix completion via convex optimization. *Communications of the ACM*, 55 (6):111–119, 2012.

Cao, Y., Liu, H., Arora, A., Augenstein, I., Röttger, P., and Hershcovich, D. Specializing large language models to simulate survey response distributions for global populations. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3141–3154, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Chen, W. and Bayati, M. Learning to recommend using non-uniform data. *arXiv preprint arXiv:2110.11248*, 2021.

Cho, S., Kim, J., and Kim, J. H. Llm-based doppelgänger models: Leveraging synthetic data for human-like responses in survey simulations. *IEEE Access*, 12:178917–178927, 2024.

Doudchenko, N. and Imbens, G. W. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. Technical Report 22791, National Bureau of Economic Research, 2016.

Du, S. S., Wang, Y., and Singh, A. On the power of truncated SVD for general high-rank matrix estimation problems. *Advances in Neural Information Processing Systems*, 30, 2017.

Duan, J., Pelger, M., and Xiong, R. Target PCA: Transfer learning large dimensional panel data. *Journal of Econometrics*, 244(2):105521, 2024.

Epstein, J. M. and Axtell, R. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Institution Press, 1996.

Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., and Li, Y. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24, 2024.

Gao, Y., Lee, D., Burtch, G., and Fazelpour, S. Take caution in using llms as human surrogates. *Proceedings of the National Academy of Sciences*, 122(24):e2501660122, 2025.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hamidi, N. and Bayati, M. On low-rank trace regression under general sampling distribution. *Journal of Machine Learning Research*, 23(321):1–49, 2022.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5 (4), December 2015. ISSN 2160-6455.

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16 (1):3367–3402, 2015.

Hollingsworth, A. and Wing, C. Tactics for design and inference in synthetic control studies: An applied example using high-dimensional data. Available at SSRN 3592088., 2020.

Horton, J. J. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

Huang, C., Wu, Y., and Wang, K. How many human survey respondents is a large language model worth? an uncertainty quantification perspective. *arXiv preprint arXiv:2502.17773*, 2025.

Hullman, J., Broska, D., Sun, H., and Shaw, A. This human study did not involve human subjects: Validating llm simulations as behavioral evidence. *arXiv preprint arXiv:2602.15785*, 2026.

Iyengar, G., Lin, Y.-S. W., and Wang, K. Model-free assessment of simulator fidelity via quantile curves, 2026.

Kolluri, A., Wu, S., Park, J. S., and Bernstein, M. S. Finetuning llms for human behavior prediction in social science experiments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30084–30099, 2025.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Laird, J. E., Newell, A., and Rosenbloom, P. S. Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64, 1987.

Leng, Y., Sang, Y., and Agarwal, A. Reduce disparity between llms and humans: Optimal llm sample calibration. *SSRN Working Paper 4802019*, 2024.

Li, A., Chen, H., Namkoong, H., and Peng, T. Llm generated persona is a promise with a catch. *arXiv preprint arXiv:2503.16527*, 2025.

Lin, Y., Chen, L., Ali, A., Nugent, C., Cleland, I., Li, R., Ding, J., and Ning, H. Human digital twin: A survey. *Journal of Cloud Computing*, 13(1):131, 2024.

Lord, F. M. *Applications of Item Response Theory to Practical Testing Problems*. Routledge, 2012.

Ludwig, J., Mullainathan, S., and Rambachan, A. Large language models: An applied econometric framework. *Annual Review of Economics*, 18, 2024.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11: 2287–2322, 2010.

McFadden, D. Conditional logit analysis of qualitative choice behavior. In Zarembka, P. (ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press, New York, 1974.

Nguyen, T. N. Toward human digital twins for cybersecurity simulations on the metaverse: Ontological and network science approach. *JMIRx Med*, 3(2):e33502, 2022.

Orlikowski, M., Pei, J., Röttger, P., Cimiano, P., Jurgens, D., and Hovy, D. Beyond demographics: Fine-tuning large language models to predict individuals' subjective text perceptions. *arXiv preprint arXiv:2502.20897*, 2025.

Park, J. S., Zou, C. Q., Shaw, A., Hill, B. M., Cai, C., Morris, M. R., Willer, R., Liang, P., and Bernstein, M. S. Generative agent simulations of 1,000 people, 2024.

Peng, T., Gui, G., Brucks, M., Merlau, D. J., Fan, G. J., Sliman, M. B., Johnson, E. J., Althenayyan, A., Bellezza, S., Donati, D., Fong, H., Friedman, E., Guevara, A., Hussein, M., Jerath, K., Kogut, B., Kumar, A., Lane, K., Li, H., Morwitz, V., Netzer, O., Perkowski, P., and Toubia, O. Digital twins as funhouse mirrors: Five key distortions, 2026.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Rossi, L., Harrison, K., and Shklovski, I. The problems of llm-generated data in social science research. *Sociologica: International Journal for Sociological Debate*, 18(2):145–168, 2024.

Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., and Hashimoto, T. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pp. 29971–30004. PMLR, 2023.

Scherrer, N., Sh, C., Feder, A., and Blei, D. M. Evaluating the moral beliefs encoded in llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Silver, N. C. and Dunlap, W. P. Averaging correlation coefficients: Should fisher's z transformation be used? *Journal of Applied Psychology*, 72(1):146–148, 1987.

Suh, J., Jahanparast, E., Moon, S., Kang, M., and Chang, S. Language model fine-tuning on scaled survey data for predicting distributions of public opinions. *arXiv preprint arXiv:2502.16761*, 2025.

Tang, Z., Chen, W., and Xu, K. Match made with matrix completion: Efficient learning under matching interference. *arXiv preprint arXiv:2601.06982*, 2026.

Toubia, O., Gui, G. Z., Peng, T., Merlau, D. J., Li, A., and Chen, H. Database report: Twin-2k-500: A data set for building digital twins of over 2,000 people based on their answers to over 500 questions. *Marketing Science*, 2025.

Train, K. E. *Discrete Choice Methods with Simulation*. Cambridge university press, 2009.

Tranchero, M., Brenninkmeijer, C.-F., Murugan, A., and Nagaraj, A. Theorizing with large language models. Working Paper 33033, National Bureau of Economic Research, October 2024.

Vershynin, R. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

Wang, B., Khoo, Z.-Y., and Wang, J. Prompts to proxies: Emulating human preferences via a compact llm ensemble. *arXiv preprint arXiv:2509.11311*, 2026a.

Wang, M., Zhang, D. J., and Zhang, H. Large language models for market research: A data-augmentation approach. *Marketing Science*, 2026b.

Xiong, R. and Pelger, M. Large dimensional latent factor modeling with missing observations and applications to causal inference. *Journal of Econometrics*, 233(1):271–301, 2023.

Xu, K., Zhao, X., Bastani, H., and Bastani, O. Group-sparse matrix factorization for transfer learning of word embeddings. In *International Conference on Machine Learning*, pp. 11603–11612. PMLR, 2021.

A. Extended Related Work

Human digital twins. The concept of a *digital twin*—a virtual representation of a physical entity or system—originated in engineering and manufacturing and has since expanded to a wide range of applied domains. A particularly important recent extension is the concept of *human digital twins*, which aims to construct faithful digital representations of human behavior and decision-making (Nguyen, 2022; Lin et al., 2024). In consumer and market research, digital twins have emerged as AI-driven behavioral simulators of customers, driven by their speed, flexibility, and cost-effectiveness relative to traditional survey- and panel-based data collection (Ludwig et al., 2024; Toubia et al., 2025; Wang et al., 2026b).

LLMs as human behavior simulators. A rapidly growing line of work has explored LLMs as proxies for humans in surveys, experiments, and agent-based simulations, documenting both promises and limitations of this paradigm (Aher et al., 2023; Horton, 2023; Santurkar et al., 2023; Scherrer et al., 2023; Gao et al., 2024; Rossi et al., 2024; Tranchero et al., 2024; Binz et al., 2025; Cao et al., 2025; Gao et al., 2025; Li et al., 2025; Hullman et al., 2026; Peng et al., 2026). Much of this literature focuses on constructing LLM-based simulators, showcasing downstream applications, or characterizing systematic failure modes such as distributional concentration, ideological biases, and sensitivity to prompt design. While these studies provide valuable insights into the capabilities and limitations of LLM-based simulation, they generally do not offer a versatile mechanism for systematically correcting the misalignment.

Closing the sim-to-real gap. To improve simulation fidelity, several lines of work have emerged. One approach is task-specific fine-tuning, which re-trains the LLM on domain-specific human response data (Cho et al., 2024; Binz et al., 2025; Cao et al., 2025; Orlikowski et al., 2025; Suh et al., 2025). While effective when sufficient training data is available, fine-tuning is computationally expensive and tightly coupled to the training distribution. A complementary line of work addresses the gap at the distributional level by reweighting LLM-generated samples to match observed human response distributions (Leng et al., 2024; Bui et al., 2025; Wang et al., 2026a). These methods offer a lightweight alternative to fine-tuning for distributional calibration, but are not designed for individual-level calibration. A related and notable line of work develops inference procedures that account for distributional misalignment between LLM and human responses (Huang et al., 2025; Iyengar et al., 2026). Our framework is complementary to all of these efforts: it provides a principled post-hoc calibration layer that can be applied on top of any simulator—zero-shot, prompt-engineered, or fine-tuned—and is designed to be data-efficient, generalizable across questions and populations, and amenable to theoretical analysis. Moreover, as we show in Section 5, existing distributional reweighting methods can be understood as special cases within our latent factor framework, which provides theoretical guarantees for their generalization to new questions.

A note on "calibration." We emphasize that *calibration* in our context refers to reducing the discrepancy between simulated and real human responses, i.e., improving simulation fidelity. This is distinct from *confidence calibration* in the machine learning literature, where methods such as Platt scaling (Platt, 1999) and temperature scaling (Guo et al., 2017) adjust a classifier's output probabilities so that predicted confidence levels match empirical accuracy. The goal here is to correct systematic biases in the *content* of simulated responses, not in a model's self-assessed confidence.

Synthetic control and matrix completion. Our framework draws on the synthetic control literature, which traditionally constructs weighted combinations of control units to approximate counterfactual outcomes for treated units in causal inference settings (Abadie & Gardeazabal, 2003; Abadie et al., 2010; Abadie, 2021) and has more broadly been recast as a tool for experimental design (Abadie & Zhao, 2021), and on matrix completion and latent factor methods for panel data (Candes & Recht, 2012; Athey et al., 2021; Xiong & Pelger, 2023), including transfer-learning extensions that use auxiliary panels to improve factor recovery (Duan et al., 2024). Several works in these areas are especially relevant. Agarwal et al. (2025) introduces *synthetic interventions*, which apply matrix-completion ideas to predict counterfactual outcomes by learning transferable structure across units—a perspective closely related to ours, though our focus is on calibrating AI-generated simulations rather than estimating causal effects. Agarwal et al. (2023) further develops causal matrix completion with theoretical guarantees under latent factor models. On the algorithmic side, Mazumder et al. (2010) proposes SoftImpute for spectral regularization-based matrix completion, and Hastie et al. (2015) develops fast alternating least squares methods for low-rank matrix recovery. Recent theoretical advances further show that low-rank methods remain effective under nonuniform observation patterns (Chen & Bayati, 2021; Hamidi & Bayati, 2022) and dependent sampling (Tang et al., 2026), and that truncated SVD provides strong guarantees even beyond exactly low-rank settings (Du et al., 2017). Conceptually, our use of digital-twin responses as an auxiliary domain also parallels group-sparse transfer in low-rank factorization, where most latent structure is shared across domains and only a sparse component requires recalibration (Xu

et al., 2021). Our framework repurposes these classical tools for the distinct goal of correcting systematic misalignment in digital-twin simulations, and our comprehensive empirical study identifies which among these methods are most effective in this new application domain.

B. Framework and Method Details

B.1. Notation

Table 5. Summary of notation.

Symbol	Description
n	Number of individuals (users)
m	Number of reference questions (items)
q_j	Question j, for $j \in [m+1]$
K	Number of response categories (distributional setting)
d	Latent dimension
$Y, \tilde{Y} \in \mathbb{R}^{n\times(m+1)}$	Real (human) and digital-twin (DT) response matrices
$Y_{1:m}, \tilde{Y}_{1:m} \in \mathbb{R}^{n\times m}$	Reference submatrices (first m columns)
$Y_{m+1} \in \mathbb{R}^n$	Unobserved real responses to new question q_{m+1}
$\tilde{Y}_{m+1} \in \mathbb{R}^n$	DT responses to new question q_{m+1}
$\hat{Y}_{m+1} \in \mathbb{R}^n$	Calibrated prediction of Y_{m+1}
$Y_v, \tilde{Y}_v$	Shorthand for $Y_{m+1}, \tilde{Y}_{m+1}$ used in Section 3, where $Y, \tilde{Y}$ also abbreviate $Y_{1:m}, \tilde{Y}_{1:m}$
$u_i, \tilde{u}_i \in \mathbb{R}^d$	Latent user embeddings (human, DT)
$v_j, \tilde{v}_j \in \mathbb{R}^d$	Latent question embeddings (human, DT)
$U, \tilde{U} \in \mathbb{R}^{n\times d}$	User embedding matrices (human, DT)
$V, \tilde{V} \in \mathbb{R}^{m\times d}$	Question embedding matrices (human, DT)
$\mathcal{E}, \tilde{\mathcal{E}}$	Noise matrices (human, DT)
P_j	Marginal distribution of human responses to q_j
$\hat{P}_j$	Calibrated distributional prediction for q_j
(w, π)	Ensemble weights (personas, dummy twins)

B.2. Detailed Calibration Methods

We provide detailed descriptions of the ten calibration methods, complementing the condensed summary and algorithm boxes (Algorithms 1 and 2) in Section 2.2. The methods span two paradigms:

(a) Fit-and-transfer (Algorithm 1). This paradigm fits a predictive model f on the DT system—using DT responses to existing questions ($\tilde{Y}_{1:m}$) to predict those to the new question ($\tilde{Y}_{m+1}$)—and transfers the fitted model to the human system. We consider six instantiations:

- **Ridge**: ℓ_2-penalized linear regression, which shrinks all coefficients uniformly toward zero (Ben-Michael et al., 2021).
- **Lasso**: ℓ_1-penalized linear regression, which encourages sparse coefficients (Hollingsworth & Wing, 2020).
- **Elastic Net (EN)**: A convex combination of ℓ_1 and ℓ_2 penalties, balancing sparsity and stability (Doudchenko & Imbens, 2016).
- **Neural Network (NN)**: A feedforward network with a single hidden layer and ReLU activations, allowing nonlinear transfer mappings.
- **Synthetic Control (SC)**: The classical synthetic control method (Abadie et al., 2010), which constrains the regression weights to lie on the probability simplex, ensuring that the predicted response is a convex combination of responses to existing questions.
- **Synthetic Intervention (SI)**: A linear mapping in the singular-value space of $\tilde{Y}$ (Agarwal et al., 2025), which computes the SVD of the DT response matrix, projects both DT and human responses onto the leading singular directions, and transfers the learned relationship.

(b) Direct matrix completion (Algorithm 2). This paradigm vertically stacks Y and $\tilde{Y}$ and directly imputes the missing half-column via matrix completion algorithms, without an explicit fit-and-transfer step. We consider four methods:

- **Hard SVD Impute (HSV)**: Iterative hard-thresholding SVD, which alternates between imputing missing entries and computing a rank-constrained SVD approximation (Mazumder et al., 2010).
- **Soft SVD Impute (SSV)**: Replaces hard thresholding with nuclear-norm regularization, yielding a convex relaxation (Mazumder et al., 2010).
- **Alternating Least Squares (ALS)**: Alternates between fixing user factors and solving for question factors, and vice versa, to find a low-rank factorization of the partially observed matrix (Hastie et al., 2015).
- **Synthetic Prior (SP)**: Uses the DT prediction $\tilde{Y}_{m+1}$ as a warm start for hard SVD impute on the human data Y alone, without leveraging the stacked structure. Among the matrix completion methods, SP is the closest analogue to standard matrix completion on real data, which, however, cannot be applied without the DT warm start since no real responses exist for the new question.

In both paradigms, de-normalization uses the DT column moments as a proxy since the true human moments for q_{m+1} are unobserved. An analysis of de-normalization bias is in Appendix F.5.

Tables 6 and 7 report the hyperparameters used for each calibration method on MovieLens and Twin-2K-500, respectively, for both the new-question (column) and new-user (row) prediction tasks. For the fit-and-transfer methods, the regularization multiplier scales the penalty term; for the matrix completion methods, rank and λ control the low-rank factorization. All neural networks use ReLU activations, Adam optimizer with learning rate 10^{-3}, batch size 128, and early stopping with patience 20.

All hyperparameters were selected heuristically based on reasonable defaults for each method class (e.g., standard regularization strengths, ranks guided by SVD diagnostics) and held fixed across all leave-one-out folds. No per-fold cross-validation or oracle tuning was performed. This is a deliberate design choice: the reported results reflect what a practitioner would obtain by picking sensible defaults without hyperparameter optimization, rather than tuned upper bounds on achievable performance. The fact that calibration consistently and substantially improves over uncalibrated baselines under these fixed, non-optimized hyperparameters demonstrates the robustness of the framework to hyperparameter choice. Systematic tuning via cross-validation would likely yield further gains.

Table 6. Hyperparameters for MovieLens.

Method	**New question**	**New user**
Ridge	$\lambda = 1000$	$\lambda = 1000$
Lasso	$\alpha = 0.01$	$\alpha = 0.1$
Elastic net	$\alpha = 0.1$, ℓ_1 ratio $= 0.1$	$\alpha = 1$, ℓ_1 ratio $= 0.1$
Synthetic control	$\lambda = 10^{-8}$	$\lambda = 1$
Neural network	hidden $= [16]$, weight_decay $= 0.1$ epochs $= 200$	hidden $= [8]$, weight_decay $= 1$ epochs $= 200$
Synthetic intervention	rank $= 50$, $\lambda = 100$	rank $= 10$, $\lambda = 100$
Hard SVD impute	rank $= 5$	rank $= 2$
Soft SVD impute	rank $= 15$, $\lambda = 5$	rank $= 5$, $\lambda = 20$
ALS	rank $= 15$, $\lambda = 5$	rank $= 3$, $\lambda = 5$
Synthetic prior	rank $= 8$	rank $= 2$

Table 7. Hyperparameters for Twin-2K-500.

Method	New question	New user
Ridge	$\lambda = 100$	$\lambda = 5000$
Lasso	$\alpha = 0.001$	$\alpha = 1$
Elastic net	$\alpha = 0.01$, ℓ_1 ratio $= 0.3$	$\alpha = 1$, ℓ_1 ratio $= 0.1$
Synthetic control	$\lambda = 10^{-6}$	$\lambda = 10^{-6}$
Neural network	hidden $= [8]$, weight_decay $= 0.05$ epochs $= 200$	hidden $= [8, 8]$, weight_decay $= 0.001$ epochs $= 200$
Synthetic intervention	rank $= 20$, $\lambda = 100$	rank $= 30$, $\lambda = 100$
Hard SVD impute	rank $= 5$	rank $= 2$
Soft SVD impute	rank $= 20$, $\lambda = 20$	rank $= 10$, $\lambda = 10$
ALS	rank $= 20$, $\lambda = 20$	rank $= 15$, $\lambda = 0.5$
Synthetic prior	rank $= 8$	rank $= 2$

C. Experimental Setup Details

C.1. MovieLens Prompt Templates

The following prompt templates are used to obtain digital-twin ratings on MovieLens (Section 2.3).

System Prompt

You, AI, are an expert in predicting human movie ratings.
You are given a user's history of movie ratings and a new movie for the same user to rate.
You are also given a format instruction that specifies the type of rating you need to provide.
You need to rate the movie as the user would rate it, based on the user's history of ratings and the format instruction.

User Prompt

PERSONA INFORMATION: `{persona_ratings}`
IN-CONTEXT EXAMPLES: `{in_context_ratings}`
NEW MOVIE: `{movie}`; GENRE: `{genre}`; TOP 10 TAGS: `{top_10_tags}`.
FORMAT INSTRUCTIONS: Only a number on a 5-star rating scale, with half-star increments (0.5–5.0). Larger numbers indicate higher ratings.

C.2. Persona Constructions for Twin-2K-500

A few persona constructions have fewer than 2,058 simulated personas due to filtering for valid responses. For those constructions, we use the subset of n participants for which both human and DT responses are available. For exposition simplicity, we refer to the number of participants as $n = 2{,}058$ throughout the paper.

The thirteen constructions vary along four axes—LLM backbone, persona format, prompting strategy, and persona content. Exact prompt templates, persona encoding schemes, and API settings are documented in Toubia et al. (2025). The thirteen constructions (all using temperature 0 unless otherwise noted) are:

1. **Text, GPT-4.1-mini** (*default*): Full survey responses provided as free-text, simulated with GPT-4.1-mini.
2. **Text, Gemini-Flash-2.5**: Same free-text persona, simulated with Gemini-Flash-2.5 to compare model-dependent fidelity.
3. **JSON, GPT-4.1-mini**: Survey responses encoded as structured JSON fields to assess the impact of input format.
4. **JSON, GPT-4.1**: Same JSON input, using the full GPT-4.1 model to evaluate the effect of increased model capacity.

5. **Text + CoT, GPT-4.1-mini**: Text persona with chain-of-thought reasoning instructions.

6. **Text + repeat, GPT-4.1-mini**: Model prompted to repeat each question and answer choice before responding, ensuring full context is processed.

7. **Text, temp=0.7, GPT-4.1-mini**: Same text input with default sampling temperature (0.7) instead of 0 to evaluate the impact of generation randomness.

8. **JSON + PO, GPT-4.1-mini**: JSON persona using OpenAI's Predicted Output feature for efficient structured output generation.

9. **JSON + PO, GPT-4.1**: Same as above with the full GPT-4.1 model.

10. **Fine-tuned GPT-4.1-mini**: GPT-4.1-mini fine-tuned on 500 labeled personas.

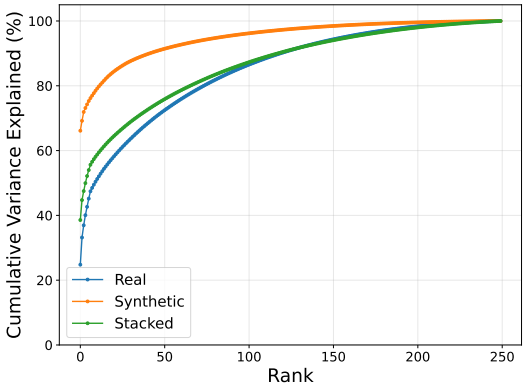
11. **Demographics only, GPT-4.1-mini**: Persona includes only the 14 demographic variables (region, sex, age, education, race, citizenship, marital status, religion, religious attendance, political party, household income, political ideology, household size, employment status).

12. **Summary, GPT-4.1-mini**: A concise persona summary instead of complete survey responses.

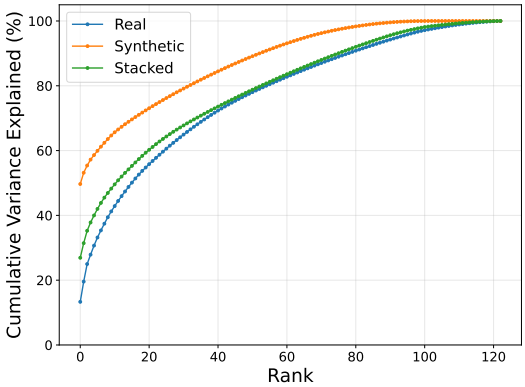
13. **Summary + JSON, GPT-4.1-mini**: Structured JSON persona augmented with an appended summary to test whether hybrid input improves results.

C.3. SVD and Row Space Diagnostics

Before SVD is applied, we demean the matrix's columns to avoid an intercept with large magnitude.



(a) MovieLens (zero-shot DT baseline).



(b) Twin-2K-500 (default: text, GPT-4.1-mini).

Figure 4. SVD diagnostics (cumulative variance explained). Both the real human response matrix Y and the DT response matrix $\tilde{Y}$ exhibit strong low-rank structure on both datasets, with a small number of singular values explaining a large fraction of the variance.

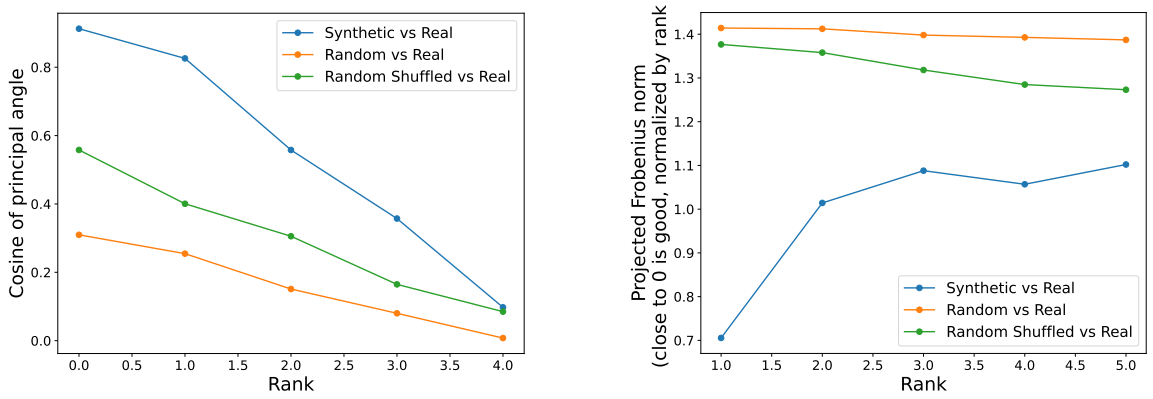


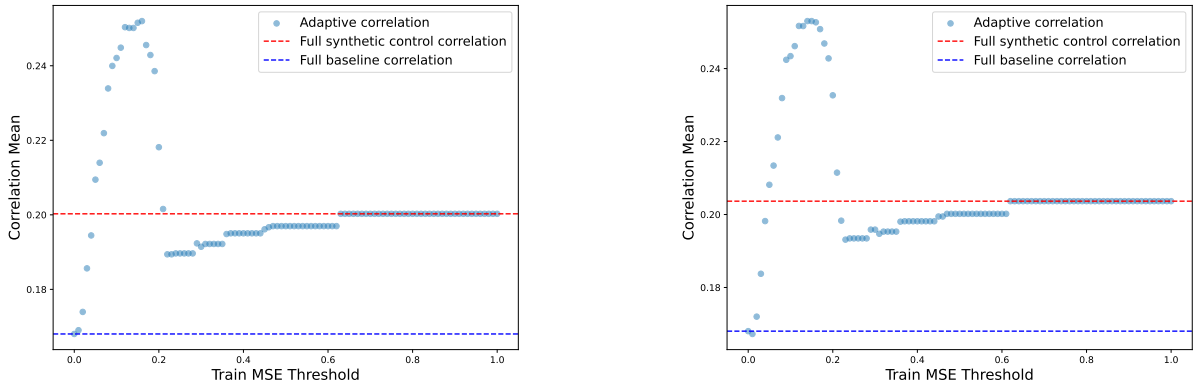
Figure 5. Row space alignment on Twin-2K-500 (default persona construction: text, GPT-4.1-mini). Cosine of principal angles (left) and projection-Frobenius norm (right) between the row spaces of $V^\top$ and $\tilde{V}^\top$, restricted to the leading $r_{\max} = r + 2$ singular directions. The DT row space aligns closely with the human row space, matching the pattern observed on MovieLens (Figure 2).

D. Extensions

D.1. Adaptive Transfer

Algorithm 1 always transfers the fitted model from DT data to human data. However, if the model does not predict the target question well on the DT system itself, then transferring it blindly may be counterproductive. This motivates a simple safeguard: use a *fit diagnostic on the synthetic system* to decide whether calibration should be applied.

Specifically, after fitting the regression of $\tilde{Y}_v$ on $\tilde{Y}$ (Step 2 of Algorithm 1), we compute the training mean-squared error (MSE) on the synthetic data. If the training MSE is below a threshold τ, we use the calibrated predictor $\hat{Y}_v = Y\hat{\beta}$; otherwise, we revert to the uncalibrated DT prediction for that target question. More generally, one could use cross-validated error on the synthetic system or other diagnostics of fit quality.



(a) Ridge *(b)* Elastic Net

Figure 6. Adaptive transfer on Twin-2K-500 (default persona construction) with Ridge (left) and EN (right). Average Pearson correlation on held-out questions as a function of the training MSE threshold τ. An optimal threshold of approximately 0.15 achieves a correlation of 0.252 for both methods, representing a 50% improvement over the DT baseline—compared to 19% (Ridge) and 21% (EN) for always-transfer calibration.

Figure 6 illustrates this adaptive rule on Twin-2K-500 (default construction (Text, GPT-4.1-mini)) with Ridge and EN. Varying the threshold τ traces a clear performance trade-off: with an oracle-optimal threshold around 0.15, adaptive transfer achieves a correlation of 0.252 for both methods—a 50% improvement over the DT baseline, compared to 19% (Ridge) and 21% (EN) for naïve transfer that always applies calibration. This substantial gain arises because certain questions are not well-situated in the linear span of the source questions; for these, the synthetic regression has high training MSE, signaling that transfer is unreliable. By reverting to the uncalibrated DT prediction in such cases, we avoid harmful calibration and

improve overall robustness.

We present this analysis as an illustration of the *potential* of adaptive transfer rather than a fully specified pipeline: the oracle threshold demonstrates how much performance can be recovered by selectively applying calibration, motivating future work on principled threshold selection strategies (e.g., cross-validated error on the synthetic system or question-specific confidence intervals).

D.2. Predicting Responses for New Users

This section complements the new-user results summarized in Section 4.2 (Table 3) with the supporting diagnostics and the deeper analysis deferred from the main text. Recall that the new-user task applies Algorithm 1 to the *transposed* response matrices; the error analysis of Theorem 3.1 carries over verbatim, with the sufficient condition for exact transfer becoming the column space inclusion (5). Beyond the Pearson correlations in Table 3, results under additional metrics (RMSE, MAE, Spearman's ρ) are reported in Appendix F.1.

Column space alignment. Empirical diagnostics confirm that condition (5) holds approximately. Figures 7 and 8 show that on both MovieLens and Twin-2K-500 (default construction (Text, GPT-4.1-mini)), the DT column spaces are substantially more aligned with the human column spaces than random baselines, mirroring the row space alignment results in Section 3.2.

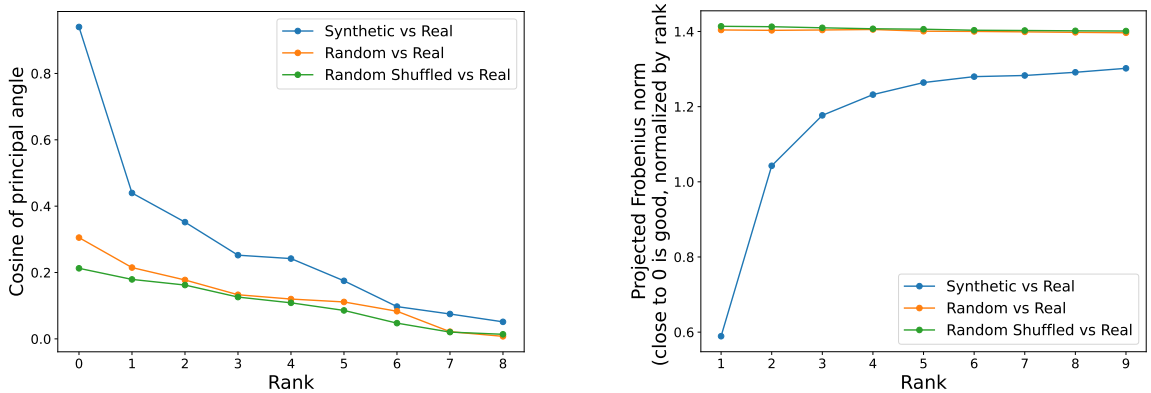


Figure 7. Column space alignment on MovieLens (zero-shot DT baseline). Cosine of principal angles (left) and projection-Frobenius norm (right) between the column spaces of U and $\tilde{U}$, restricted to the leading $r_{\max} = r + 2$ singular directions. The DT column space aligns closely with the human column space, supporting the applicability of our framework to new-user prediction.

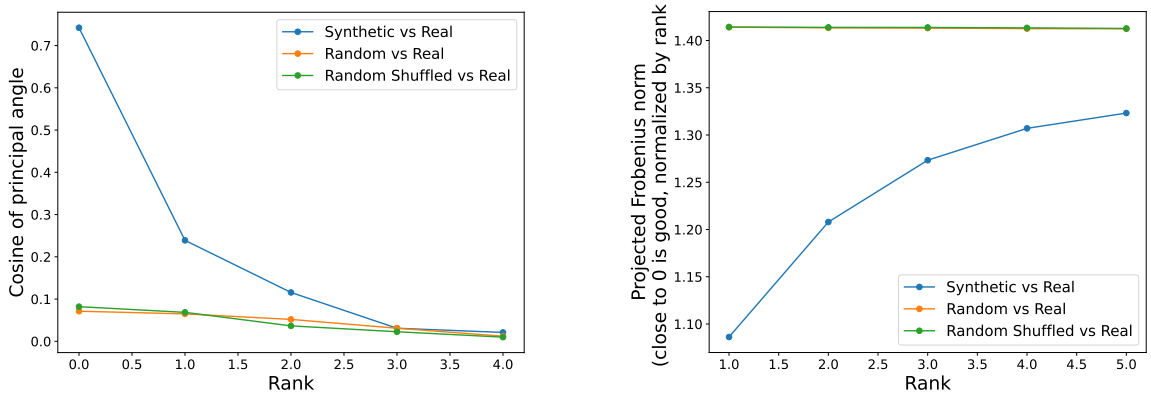


Figure 8. Column space alignment on Twin-2K-500 (default persona construction: text, GPT-4.1-mini). Cosine of principal angles (left) and projection-Frobenius norm (right) between the column spaces of U and $\tilde{U}$, restricted to the leading $r_{\max} = r + 2$ singular directions.

Nonlinearity in user space. Unlike the new-question setting where elastic net dominates, neural network (NN) is the best method on both datasets for new-user prediction. This suggests that cross-user relationships exhibit more nonlinearity than

cross-question relationships. One possible explanation is that user heterogeneity is inherently higher-dimensional: while question structure may follow a few dominant latent factors (e.g., genre, difficulty), user preferences involve more complex interactions that benefit from a nonlinear transfer mapping.

Compressed spread across methods. The performance spread across methods is dramatically narrower for new-user prediction: correlations range from .361 to .377 on MovieLens (a spread of .016) and .863 to .877 on Twin-2K-500 (a spread of .014), compared to .454 to .524 (spread of .070) for new-question prediction on MovieLens and .121 to .204 (spread of .083) on Twin-2K-500 with the same default construction. This compressed spread can be attributed to the high redundancy of the feature space in new-user prediction. In the transposed formulation, the "features" are existing users ($n-1$ of them), while "observations" are questions (m). Since $n \gg d$ (the number of users far exceeds the latent rank), the dictionary of users available to represent any new user is massively over-complete: many near-equivalent combinations of existing users span the same low-dimensional signal. Consequently, different methods—whether sparse or dense, linear or nonlinear—select different subsets or weightings but project onto essentially the same latent directions. In contrast, for new-question prediction, the features are questions (m), and the lower ratio m/d means each question carries more unique information, so the specific choice of regularization and weighting has a larger impact on which aspects of the latent structure are captured.

E. Distributional Calibration Details

E.1. Assumptions

Assumption E.1 (Linear probability model)**.** There exist human user embeddings $u_i = (u_{i,1}, \ldots, u_{i,K})$, digital twin embeddings $\tilde{u}_i = (\tilde{u}_{i,1}, \ldots, \tilde{u}_{i,K})$, and *shared* question embeddings $v_j = (v_{j,1}, \ldots, v_{j,K})$ such that for each user i, question j, and response category $k \in [K]$,

$$\Pr(Y_{ij} = k \mid u_i) = \langle u_{i,k},\, v_{j,k} \rangle,$$

$$\Pr(\tilde{Y}_{ij} = k \mid \tilde{u}_i) = \langle \tilde{u}_{i,k},\, v_{j,k} \rangle,$$

where $u_{i,k}, \tilde{u}_{i,k}, v_{j,k} \in \mathbb{R}^d$.

Assumption E.2 (User embeddings)**.** The human user embeddings $\{u_i\}_{i=1}^n$ are i.i.d. from μ, and the digital twin embeddings $\{\tilde{u}_i\}_{i=1}^n$ are i.i.d. from $\tilde{\mu}$, where $\tilde{\mu}$ is supported on a finite set $\mathcal{U}$.

Assumption E.3 (Span of question embeddings)**.** There exist coefficients $c_1, \ldots, c_m \geq 0$ with $\sum_{j=1}^m c_j = 1$ such that for all $k \in [K]$, $v_{m+1,k} = \sum_{j=1}^m c_j v_{j,k}$.

E.2. Algorithm

Algorithm 3 Distributional calibration via weighted ensemble

Require: Training distributions $\{P_j\}_{j=1}^m$, DT responses $\{\tilde{Y}_{ij}\}_{i\in[n],\, j\in[m+1]}$, response categories $[K]$, discrepancy measure D

Ensure: Calibrated distributional prediction $\hat{P}_{m+1}$

1: Fit weights $(\hat{w}, \hat{\pi})$ by solving (7) via mirror descent.

2: Predict $\hat{P}_{m+1}$ by (8).

E.3. Prompts

The following prompt templates are used for the distributional calibration experiments (Section 5.3).

System Prompt (MovieLens)

You, AI, are an expert in predicting human preferences. You are given a persona profile and a movie to rate, and also a format instruction that specifies the type of rating you need to provide. You need to rate the movie as the persona would rate it, based on the persona profile and the format instructions.

User Prompt (MovieLens)

```
{persona}
MOVIE: {movie}
GENRE: {genre}
TOP 10 TAGS: {top_10_tags}
```
FORMAT INSTRUCTIONS: Only a number on a 5-star rating scale, with half-star increments (0.5 - 5.0). Larger numbers indicate higher ratings.

System Prompt (OpinionQA)

You, AI, are an expert in predicting human answers to survey questions. You are given a persona profile and a survey question to answer, and also a format instruction that specifies the type of answer you need to provide. You need to answer the survey question as the persona would answer it, based on the persona profile and the format instructions.

User Prompt (OpinionQA)

```
{persona}
SURVEY QUESTION: {question}
```
FORMAT INSTRUCTIONS: Only an integer number from 1 to 5, corresponding to the choice of answer to the survey question.

E.4. Discrepancy Measures

Let P and Q be two probability distributions over $[K]$. We evaluate distributional calibration using the following discrepancy measures.

- **Total variation distance:** $\mathrm{TV}(P, Q) = \frac{1}{2} \sum_{k=1}^{K} |P(k) - Q(k)|$.
- χ^2 **divergence:** $D_{\chi^2}(P\|Q) = \sum_{k=1}^{K} \frac{P(k)^2}{Q(k)} - 1$.
- **Kullback–Leibler (KL) divergence:** $D_{\mathrm{KL}}(P\|Q) = \sum_{k=1}^{K} P(k) \log \frac{P(k)}{Q(k)}$.
- **Hellinger distance:** $H^2(P, Q) = 1 - \sum_{k=1}^{K} \sqrt{P(k)Q(k)}$.

Let F and G be the cumulative distribution functions (CDFs) of P and Q, respectively. We define:

- **Kolmogorov–Smirnov (KS) distance:** $D_{\mathrm{KS}}(P, Q) = \|F - G\|_\infty$.
- **CDF** ℓ_1 **distance:** $\|F - G\|_1$.
- **CDF** ℓ_2 **distance:** $\|F - G\|_2^2$.

E.5. Variance Retention and Role of Dummy Twins

Role of dummy twins. Using both personas and dummy twins consistently achieves the best performance across both datasets. The benefit of dummy twins is most pronounced for metrics that penalize missing support (χ^2, KL), where the personas-only variant can produce extremely large values because the DT ensemble may not cover all response categories. The K dummy twins guarantee full support by construction. The dummies-only variant substantially underperforms, confirming that DT personas carry meaningful distributional information beyond support coverage.

Variance retention. The calibrated ensemble achieves variance ratios (predicted/true) closer to 1 compared to the baseline, indicating that calibration preserves the natural variability of response distributions (Figure 9).

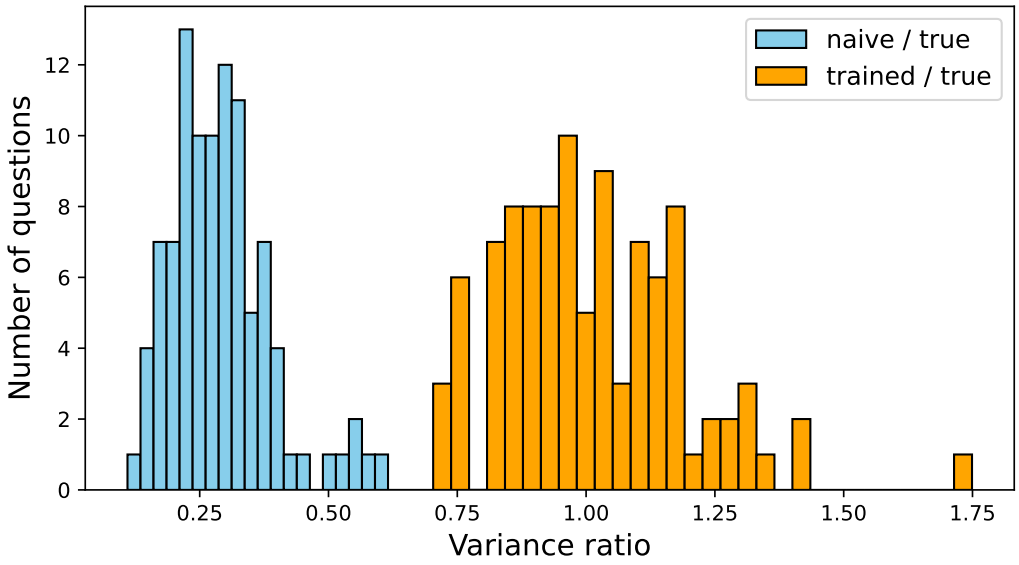


Figure 9. Histogram of variance ratios (predicted variance / true variance) on MovieLens (trained with ℓ_1 loss). The calibrated ensemble (orange) achieves variance ratios closer to 1 compared to the baseline (blue), indicating better variance retention.

E.6. Sensitivity to Training Objective and Scalability

Sensitivity to training objective. The full cross-metric tables (Tables 8 and 9) reveal that the choice of training objective D has a meaningful but bounded effect on test performance. Training with TV or KL divergence provides the most reliable cross-metric generalization on both datasets, while CDF-based objectives (ℓ_1, ℓ_2, KS) produce more variable results—in particular, χ^2 test divergence can be orders of magnitude worse when training with ℓ_1 or ℓ_2.

Scalability. Since the number of digital twins n should be large enough to adequately cover the target population, scalability in n is important. We provide evidence that the mirror descent optimization scales well with n; details are in Appendix F.4.

E.7. Full Results

Tables 8 and 9 report the full distributional prediction results on MovieLens and OpinionQA, respectively, training with every discrepancy objective and evaluating on all metrics. For each training objective, three ensemble variants are shown: personas + dummies (P+D), personas only (P), and dummies only (D).

Table 8. Full distributional prediction results on MovieLens. For each training objective, three variants are shown: personas + dummies (P+D), personas only (P), dummies only (D). Lower is better for all metrics.

Train	Variant	TV	χ^2	KL	Hellinger	KS	ℓ_1	ℓ_2	MSE
TV	P+D	.154 (.006)	.208 (.018)	.092 (.007)	.023 (.002)	.123 (.006)	.492 (.029)	.054 (.005)	.078
	P	.233 (.008)	.587 (.050)	.230 (.014)	.059 (.003)	.182 (.008)	.962 (.046)	.151 (.013)	3.61
	D	.240 (.009)	7.71 (4.56)	.263 (.027)	.053 (.004)	.168 (.008)	.534 (.027)	.072 (.007)	6.55
χ^2	P+D	.162 (.006)	.187 (.013)	.093 (.006)	.024 (.002)	.128 (.006)	.525 (.028)	.058 (.006)	.082
	P	.290 (.009)	631 (68.4)	1.44 (.073)	.132 (.006)	.199 (.008)	.674 (.033)	.107 (.011)	.096
	D	.180 (.007)	.222 (.016)	.112 (.008)	.030 (.002)	.158 (.008)	.665 (.039)	.096 (.010)	.256
KL	P+D	.156 (.006)	.195 (.015)	.091 (.006)	.023 (.002)	.124 (.006)	.498 (.028)	.054 (.005)	.077
	P	.284 (.008)	631 (68.3)	1.43 (.071)	.127 (.005)	.200 (.008)	.666 (.029)	.104 (.010)	.084
	D	.174 (.008)	.230 (.019)	.110 (.009)	.029 (.002)	.155 (.009)	.646 (.042)	.096 (.011)	.243
Hellinger	P+D	.158 (.006)	.211 (.018)	.094 (.007)	.023 (.002)	.125 (.006)	.484 (.028)	.053 (.005)	.073
	P	.284 (.008)	632 (68.2)	1.43 (.071)	.126 (.005)	.204 (.008)	.675 (.029)	.107 (.010)	.083
	D	.173 (.008)	.240 (.021)	.112 (.009)	.028 (.002)	.155 (.009)	.640 (.042)	.097 (.011)	.240
KS	P+D	.177 (.007)	13.4 (4.57)	.194 (.020)	.034 (.002)	.118 (.005)	.420 (.023)	.041 (.004)	.048
	P	.355 (.006)	.719 (.028)	.365 (.012)	.104 (.004)	.295 (.009)	1.54 (.057)	.368 (.023)	9.57
	D	.346 (.009)	67.4 (17.2)	.799 (.046)	.118 (.005)	.251 (.009)	.817 (.027)	.152 (.012)	11.7
ℓ_1	P+D	.188 (.006)	23.5 (8.10)	.212 (.021)	.036 (.002)	.123 (.005)	.415 (.022)	.041 (.004)	.043
	P	.375 (.006)	.817 (.027)	.405 (.012)	.115 (.004)	.320 (.009)	1.68 (.059)	.433 (.026)	10.9
	D	.337 (.009)	54.0 (16.7)	.679 (.044)	.108 (.005)	.251 (.009)	.769 (.029)	.150 (.012)	11.7
ℓ_2	P+D	.212 (.008)	4.05 (1.25)	.244 (.017)	.047 (.003)	.129 (.005)	.444 (.022)	.045 (.004)	.042
	P	.375 (.006)	.817 (.027)	.405 (.012)	.115 (.004)	.320 (.009)	1.68 (.059)	.433 (.026)	10.9
	D	.380 (.009)	636 (67.8)	1.71 (.062)	.171 (.005)	.279 (.010)	.921 (.030)	.189 (.014)	13.1
Baseline		.381 (.009)	28600 (3100)	2.44 (.093)	.176 (.005)	.280 (.010)	.922 (.030)	.189 (.014)	.104

Table 9. Full distributional prediction results on OpinionQA. For each training objective, three variants are shown: personas + dummies (P+D), personas only (P), dummies only (D). Lower is better for all metrics.

Train	Variant	TV	χ^2	KL	Hellinger	KS	ℓ_1	ℓ_2	MSE
TV	P+D	.171 (.009)	.255 (.030)	.114 (.011)	.030 (.003)	.150 (.009)	.284 (.019)	.046 (.006)	.101
	P	.233 (.009)	.480 (.042)	.232 (.014)	.068 (.003)	.199 (.009)	.503 (.020)	.089 (.008)	1.15
	D	.225 (.011)	.506 (.055)	.190 (.016)	.047 (.004)	.199 (.011)	.335 (.020)	.068 (.008)	2.36
χ^2	P+D	.174 (.009)	.231 (.023)	.114 (.010)	.031 (.003)	.154 (.009)	.303 (.018)	.048 (.005)	.108
	P	.298 (.015)	325 (86.5)	.864 (.109)	.106 (.010)	.267 (.015)	.459 (.027)	.127 (.014)	.241
	D	.220 (.010)	.313 (.029)	.167 (.014)	.048 (.004)	.203 (.010)	.433 (.023)	.088 (.010)	.299
KL	P+D	.174 (.009)	.247 (.027)	.115 (.010)	.031 (.003)	.154 (.009)	.290 (.018)	.046 (.006)	.101
	P	.289 (.015)	313 (85.5)	.832 (.106)	.101 (.009)	.258 (.015)	.442 (.026)	.119 (.014)	.222
	D	.216 (.010)	.328 (.035)	.164 (.015)	.046 (.004)	.202 (.011)	.421 (.024)	.088 (.010)	.314
Hellinger	P+D	.178 (.009)	.275 (.032)	.121 (.011)	.032 (.003)	.158 (.009)	.288 (.017)	.047 (.006)	.100
	P	.289 (.015)	293 (82.3)	.814 (.103)	.100 (.009)	.258 (.015)	.437 (.026)	.117 (.013)	.214
	D	.216 (.011)	.351 (.039)	.167 (.016)	.045 (.004)	.204 (.011)	.417 (.024)	.089 (.011)	.324
KS	P+D	.174 (.009)	.270 (.030)	.118 (.011)	.031 (.003)	.150 (.009)	.275 (.017)	.044 (.005)	.089
	P	.262 (.013)	1.50 (.250)	.322 (.033)	.068 (.006)	.229 (.014)	.381 (.023)	.092 (.011)	.170
	D	.348 (.016)	275 (79.2)	1.02 (.101)	.133 (.010)	.303 (.016)	.516 (.028)	.154 (.015)	5.00
ℓ_1	P+D	.178 (.009)	.298 (.036)	.123 (.012)	.031 (.003)	.154 (.009)	.271 (.017)	.045 (.006)	.087
	P	.276 (.014)	2.79 (.526)	.396 (.042)	.077 (.007)	.241 (.014)	.397 (.023)	.101 (.012)	.168
	D	.348 (.016)	275 (79.2)	1.02 (.101)	.133 (.010)	.303 (.016)	.516 (.028)	.154 (.015)	5.00
ℓ_2	P+D	.182 (.009)	.282 (.030)	.123 (.011)	.032 (.003)	.156 (.009)	.281 (.016)	.044 (.005)	.086
	P	.280 (.014)	1.62 (.212)	.362 (.031)	.075 (.006)	.231 (.013)	.380 (.020)	.089 (.009)	.204
	D	.339 (.016)	25.9 (8.23)	.710 (.060)	.118 (.008)	.295 (.015)	.501 (.027)	.145 (.014)	4.92
Baseline		.350 (.016)	11800 (3600)	1.30 (.138)	.137 (.010)	.303 (.016)	.520 (.028)	.154 (.015)	.271

F. Ablation Studies and Robustness

F.1. Supplementary Evaluation Metrics

The main text reports Pearson correlation as the primary individual-level metric. Pearson correlation is affine-invariant: it measures the strength and direction of the linear association between predicted and actual responses, regardless of location or scale shifts. While this invariance is useful for isolating structural fidelity, other metrics may capture complementary aspects of calibration quality. We therefore supplement Pearson with three additional metrics: root mean squared error (RMSE, lower is better), mean absolute error (MAE, lower is better), and Spearman's rank correlation ρ (higher is better). RMSE and MAE capture level-sensitive prediction accuracy, while Spearman captures monotonic association without assuming linearity. Together, these four metrics provide a comprehensive view of calibration quality.

New-question prediction: MovieLens. Table 10 reports all four metrics for the column-wise leave-one-out evaluation on MovieLens (250 folds). Method rankings are broadly consistent across metrics: EN is the top or near-top method on all four measures, followed by Ridge, ALS, and SI. The RMSE and MAE results confirm that calibration delivers level-sensitive improvements as well: RMSE drops from .986 (baseline) to .865–.870 (best methods), a $\approx 12\%$ reduction. Spearman closely tracks Pearson, confirming that calibration improves rank-based agreement as well.

Table 10. Supplementary evaluation metrics for new-question prediction on MovieLens (column-wise leave-one-out, 250 folds). Mean above, standard error in parentheses below. Bold denotes best.

	BL	Fit-and-transfer						Matrix Completion			
		Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
Pearson	.349	.511	.497	**.524**	.504	.457	.515	.473	.511	.512	.454
	(.005)	(.005)	(.006)	(.005)	(.005)	(.005)	(.005)	(.008)	(.005)	(.005)	(.005)
RMSE ↓	.986	.881	.882	.870	.876	.891	.873	.991	.866	**.865**	.885
	(.009)	(.009)	(.009)	(.009)	(.009)	(.009)	(.009)	(.040)	(.009)	(.009)	(.009)
MAE ↓	.725	.683	.683	.674	.678	.692	.676	.766	.671	**.670**	.686
	(.008)	(.008)	(.008)	(.008)	(.008)	(.008)	(.008)	(.031)	(.008)	(.008)	(.008)
Spearman	.358	.501	.485	**.512**	.498	.452	.503	.464	.500	.501	.450
	(.005)	(.005)	(.006)	(.005)	(.005)	(.005)	(.005)	(.008)	(.005)	(.005)	(.005)

New-question prediction: Twin-2K-500. Table 11 reports the same four metrics for the default persona construction (Text, GPT-4.1-mini) on Twin-2K-500 (column-wise leave-one-out, 123 folds). Method rankings on Twin-2K-500 are consistent with MovieLens. EN is the top method on all four metrics. RMSE drops from 4.22 (baseline) to 3.41 (EN), a $\approx 19\%$ reduction.

Table 11. Supplementary evaluation metrics for new-question prediction on Twin-2K-500 (column-wise leave-one-out, 123 folds, default spec: Text, GPT-4.1-mini). Mean above, standard error in parentheses below. Bold denotes best.

	BL	Fit-and-transfer						Matrix Completion			
		Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
Pearson	.168	.200	.195	**.204**	.198	.147	.190	.170	.202	.201	.121
	(.015)	(.021)	(.021)	(.021)	(.020)	(.013)	(.017)	(.017)	(.018)	(.018)	(.016)
RMSE ↓	4.22	3.42	3.45	**3.41**	3.46	3.61	3.52	4.59	3.52	3.52	3.89
	(.84)	(.68)	(.68)	(.68)	(.68)	(.71)	(.70)	(1.00)	(.70)	(.70)	(.78)
MAE ↓	3.23	2.58	2.60	**2.57**	2.61	2.71	2.66	3.49	2.66	2.66	2.93
	(.64)	(.50)	(.50)	(.50)	(.50)	(.53)	(.53)	(.76)	(.53)	(.53)	(.59)
Spearman	.170	.196	.193	**.201**	.195	.146	.185	.166	.197	.197	.119
	(.015)	(.020)	(.020)	(.021)	(.019)	(.013)	(.016)	(.017)	(.018)	(.018)	(.015)

New-user prediction: MovieLens and Twin-2K-500. Tables 12 and 13 report the four metrics for new-user prediction on both datasets. On MovieLens (Table 12), the method rankings on Pearson and Spearman are consistent: NN is the top method on both association measures, and all calibration methods improve substantially over the baseline. On Twin-2K-500 (Table 13), calibration consistently improves the level-sensitive metrics: RMSE decreases from 10.4 (baseline) to 8.96 (NN), a $\approx 14\%$ reduction, and MAE decreases from 3.62 to 3.22 ($\approx 11\%$ reduction). Pearson correlation also improves from .847 to .877. On Spearman's ρ, NN (.796) slightly edges out the baseline (.794), but most other calibrated methods fall below it. This is consistent with the high baseline Pearson correlation (.847) on this task: because the persona is constructed from each user's full survey responses, the uncalibrated DT already orders users' responses across questions quite accurately, leaving limited headroom for rank-based improvement.

Table 12. Supplementary evaluation metrics for new-user prediction on MovieLens (row-wise leave-one-out, 500 folds). Mean above, standard error in parentheses below. Bold denotes best.

	BL	Fit-and-transfer						Matrix Completion			
		Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
Pearson	.246	.373	.372	.373	**.377**	.375	.361	.372	.372	.372	.366
	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.007)
RMSE ↓	.961	**.881**	.882	.882	.885	.905	.887	.948	.947	.947	.883
	(.011)	(.009)	(.010)	(.010)	(.010)	(.011)	(.009)	(.012)	(.012)	(.012)	(.010)
MAE ↓	.720	**.689**	**.689**	.690	.694	.713	.696	.758	.758	.757	**.689**
	(.009)	(.008)	(.008)	(.008)	(.008)	(.009)	(.008)	(.010)	(.010)	(.010)	(.008)
Spearman	.238	.358	.359	.360	**.361**	.360	.343	.358	.358	.358	.350
	(.005)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.006)	(.007)

Table 13. Supplementary evaluation metrics for new-user prediction on Twin-2K-500 (row-wise leave-one-out, 2058 folds, default spec: Text, GPT-4.1-mini). Mean above, standard error in parentheses below. Bold denotes best.

	BL	Fit-and-transfer						Matrix Completion			
		Ridge	Lasso	EN	NN	SC	SI	HSV	SSV	ALS	SP
Pearson	.847	.875	.873	.875	**.877**	.874	.875	.873	.873	.873	.863
	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)
RMSE ↓	10.4	9.01	9.05	9.02	**8.96**	9.03	9.02	9.07	9.06	9.07	9.87
	(.07)	(.06)	(.06)	(.06)	(.06)	(.06)	(.06)	(.06)	(.06)	(.06)	(.07)
MAE ↓	3.62	3.26	3.29	3.26	**3.22**	3.27	3.25	3.29	3.29	3.29	3.53
	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)	(.02)
Spearman	.794	.786	.778	.781	**.796**	.781	.788	.777	.777	.777	.766
	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)

F.2. Null Digital Twin Diagnostic

To investigate whether calibration gains are attributable to DT-specific structure or artifacts of the calibration procedure, we replace the DT response matrix $\tilde{Y}$ with two null surrogates: (i) *random Gaussian noise* drawn i.i.d. from $\mathcal{N}(\bar{\mu}, \bar{\sigma}^2)$ matching the global DT mean and standard deviation, preserving the original missing pattern; and (ii) *column-shuffled real data*, in which the observed human entries are permuted within each column, preserving marginal distributions but destroying individual-level structure. We then run the same calibration pipeline (for new-question prediction) with each null DT and compare the resulting Pearson correlations with those obtained from the actual DT responses.

Table 14. Null DT diagnostic on MovieLens: Pearson correlation under actual DT, random Gaussian null, and column-shuffled null.

	BL	Ridge	EN	NN	SI
Actual DT	**.349**	**.511**	**.524**	**.504**	**.515**
Random Gaussian	$-.002$	$-.002$	.003	.007	$-.011$
Shuffled Real	.002	.013	.018	.030	.016

Table 15. Null DT diagnostic on Twin-2K-500 (default spec: Text, GPT-4.1-mini): Pearson correlation under actual DT, random Gaussian null, and column-shuffled null.

	BL	Ridge	EN	NN	SI
Actual DT	**.168**	**.200**	**.204**	**.198**	**.190**
Random Gaussian	$-.002$	$-.018$	$-.018$	$-.005$	.007
Shuffled Real	$-.001$	.004	.006	.005	$-.010$

Tables 14 and 15 report the results for a subset of five high-performing methods on MovieLens and Twin-2K-500 (default spec), respectively. All methods collapse to near-zero (often negative) correlation under both null DTs. This strongly indicates that their calibration gains are not artifacts of the calibration procedure or patterns in Y alone—they require genuine DT signal.

F.3. Robustness to Missing Data Treatment

The calibration pipeline uses iterative hard-thresholding SVD (hard SVD) as the default method for imputing missing values in the combined human–DT response matrix before fitting. This section examines sensitivity to this choice by comparing four imputation strategies:

1. **Hard SVD** (default): Iterative SVD with hard singular value thresholding.
2. **Soft SVD**: Iterative SVD with soft singular value thresholding (shrinkage).
3. **Column-mean**: Each missing entry is filled with the column mean of the observed entries.
4. **Zero-fill**: All missing entries are set to zero.

Tables 16 and 17 report the Pearson correlation for each imputation method on MovieLens and Twin-2K-500 (default spec), respectively, using a representative subset of six calibration methods.

Table 16. Imputation robustness on MovieLens: Pearson correlation by imputation method (column-wise leave-one-out, 250 folds). Bold denotes best within each column.

Imputation	BL	Ridge	EN	NN	SC	SI
Hard SVD (default)	.349	.511	.524	.504	.457	.515
Soft SVD	.349	.511	.524	.503	.457	.516
Column-mean	.349	**.518**	**.532**	**.516**	**.468**	**.528**
Zero-fill	.349	.398	.387	.396	.344	.380

Table 17. Imputation robustness on Twin-2K-500 (default spec: Text, GPT-4.1-mini): Pearson correlation by imputation method (column-wise leave-one-out, 123 folds). Bold denotes best within each column.

Imputation	BL	Ridge	EN	NN	SC	SI
Hard SVD (default)	.168	.200	.204	.198	.147	.190
Soft SVD	.168	.200	.203	.198	.147	.190
Column-mean	.168	**.213**	**.214**	**.212**	**.182**	**.200**
Zero-fill	.168	.203	.206	.200	.176	.139

Discussion. Hard SVD and soft SVD yield nearly identical results on both datasets, indicating that the choice between hard and soft singular value thresholding is immaterial. Column-mean imputation is slightly better than SVD-based methods across the board (e.g., EN improves from .524 to .532 on MovieLens), possibly because it exactly preserves column means,

which the subsequent normalization step relies on. Zero-fill imputation substantially degrades performance on MovieLens (EN drops from .524 to .387) and produces mixed, often inferior behavior on Twin-2K-500: zero-filling the normalized response matrix distorts the covariance structure that the calibration methods rely on.

Overall, results are robust across the three reasonable imputation choices (hard SVD, soft SVD, and column-mean). Zero-fill serves mainly as a stress test and is not competitive in general. The default choice of hard SVD is reasonable, while column-mean imputation is a viable alternative that may yield marginally better results.

F.4. Computational Cost and Scalability

This section reports wall-clock times for the full leave-one-out evaluations and analyzes the scaling behavior of the distributional calibration mirror descent algorithm. All timings were measured on a single Apple MacBook Pro (M1 chip, 8-core CPU/GPU, 16 GB memory).

Table 18 reports wall-clock times for the full leave-one-out evaluations on both datasets. MovieLens has 500×250 entries ($\approx 22\%$ missing); Twin-2K-500 has $2{,}058 \times 123$ entries ($\approx 23\%$ missing, default spec). Fit-and-transfer methods (Ridge, Lasso, EN, NN, SC, SI) have comparable runtimes that scale linearly with the number of leave-one-out folds. Several matrix completion methods are cheaper overall: MC ALS and MC Synthetic Prior are the fastest, completing column-wise evaluations in under a minute on both datasets. For row-wise evaluation on the larger Twin-2K-500 dataset, the contrast is sharpest for MC Synthetic Prior (34 s) and MC Hard SVD (77 s), while MC Soft SVD is comparable to the fit-and-transfer methods.

Table 18. Wall-clock times (seconds) for leave-one-out evaluation on MovieLens and Twin-2K-500. Fit-and-transfer methods (top) scale with the number of folds; matrix completion methods (bottom) are often faster overall. Row-wise evaluation on Twin-2K-500 is most expensive ($\approx$ 19–20 minutes for fit-and-transfer), though MC Synthetic Prior and MC Hard SVD remain fast, while MC Soft SVD is comparable to the fit-and-transfer methods.

	MovieLens (500×250)		**Twin-2K-500** ($2{,}058 \times 123$)	
Method	**Col (250 folds)**	**Row (500 folds)**	**Col (123 folds)**	**Row (2058 folds)**
Ridge	120	232	123	1160
Lasso	117	229	129	1157
Elastic Net	115	227	128	1166
Neural Net	155	243	152	1229
Synthetic Control	125	269	124	1226
Synthetic Intervention	119	235	123	1166
MC Hard SVD	81	60	56	77
MC Soft SVD	111	62	57	1152
MC ALS	48	44	49	800
MC Synthetic Prior	52	51	60	34

Mirror descent scaling. Table 19 reports the wall-clock time for distributional calibration via mirror descent as a function of the number of digital twins n, with $m = 250$ questions and $K = 5$ categories (MovieLens). At $n = 10$ the wall time is essentially fixed overhead (Python startup, data I/O). From $n = 500$ onward the runtime grows roughly linearly in n: doubling n from 500 to 1000 increases time from 8.9 s to 14.2 s ($\approx 1.6\times$), and doubling again to 2058 yields 21.6 s ($\approx 1.5\times$), both close to the ideal $2\times$ once the ≈ 5 s fixed overhead is subtracted.

Table 19. Mirror descent wall-clock time as a function of the number of digital twins n ($m = 250$, $K = 5$, MovieLens).

n	10	500	1000	2058
Time (s)	8.0	8.9	14.2	21.6

Complexity analysis. Each mirror-descent iteration recomputes the ensemble distribution $\hat{P}_j(w, \pi)$ and its gradient with respect to (w, π); both operations scale as $O(nmK)$ in the number of twins n, reference questions m, and categories K.

The number of iterations T is a fixed hyperparameter, so the total cost is $O(TnmK)$ plus $O(nm)$ data-loading overhead that dominates at very small n. The empirical wall times confirm this prediction: subtracting the ≈ 5 s overhead yields ≈ 3.9 s at $n = 500$, ≈ 9.2 s at $n = 1000$, and ≈ 16.6 s at $n = 2058$—within $\approx 15\%$ of a perfectly linear-in-n curve through $n = 2058$.

F.5. De-normalization Bias Analysis

The calibration pipeline normalizes both human and DT response matrices column-wise (subtracting the column mean, dividing by the column standard deviation) before fitting, and de-normalizes predictions using the target column's moments. Because the target column has no observed human responses, de-normalization uses the DT column moments (mean $\tilde{\mu}_j$, standard deviation $\tilde{\sigma}_j$) as proxies for the unknown human moments (μ_j, σ_j). This section quantifies the typical discrepancy between DT and human moments. For each reference column j (where both human and DT responses are observed), we compute the mean difference $\delta\mu_j = \bar{\tilde{Y}}_j - \bar{Y}_j$ and the standard deviation ratio $\tilde{\sigma}_j/\sigma_j$.

MovieLens. Table 20 summarizes the distribution of $\delta\mu$ and the standard deviation ratio across the 250 reference columns. DT means are slightly higher than human means on average ($\delta\mu \approx 0.27$ on a 1–5 star scale). DT standard deviations are systematically lower ($\approx 70\%$ of human standard deviations), confirming the well-known LLM concentration phenomenon: LLM-generated responses tend to cluster more narrowly around the mean than real human responses.

Table 20. De-normalization bias on MovieLens: distribution of moment discrepancies across 250 reference columns.

Metric	Mean	Median	Std	Min	Max	IQR
$\delta\mu$ (DT − Human)	0.265	0.228	0.293	−0.507	0.965	0.385
Std ratio (DT/Human)	0.705	0.685	0.121	0.438	1.323	0.123
Var ratio (DT/Human)	0.511	0.469	0.192	0.192	1.751	0.172

Twin-2K-500. Table 21 reports the same analysis across six representative persona constructions (123 reference columns each). The mean shift $\delta\mu$ is roughly centered near zero across specs (no systematic directional bias), but with high variance (std 3–8) driven by outlier questions with extreme value ranges. The standard deviation ratio is consistently 0.5–0.76 across all constructions, confirming that the DT concentration phenomenon is universal and not specific to any particular LLM or persona format.

Table 21. De-normalization bias on Twin-2K-500: moment discrepancies for six representative persona constructions (123 reference columns each).

Persona construction	$\delta\mu$ Mean	$\delta\mu$ Median	$\delta\mu$ Std	Std ratio mean	Std ratio median
Text, GPT-4.1-mini	−0.361	0.110	5.353	0.604	0.763
JSON, GPT-4.1	−0.209	−0.009	3.592	0.643	0.801
Text, Gemini-Flash-2.5	0.174	−0.152	3.244	0.757	0.951
Fine-tuned GPT-4.1-mini	−0.942	−0.173	3.565	0.645	0.762
Demographics only	0.508	−0.232	7.570	0.525	0.490
Summary, GPT-4.1-mini	0.153	−0.175	7.808	0.628	0.680

Discussion. Because Pearson correlation is affine-invariant, the de-normalization bias *cannot affect* the primary evaluation metric reported in the main text. It does affect level-sensitive metrics (RMSE, MAE), but the supplementary metrics in Section F.1 show that calibration improves RMSE and MAE substantially *despite* using approximate DT moments for de-normalization. A learned moment correction—for example, a linear mapping from DT column moments to human column moments estimated via leave-one-out on reference columns—is a natural extension that could further improve level-sensitive accuracy. We leave this for future work.

G. Proofs

G.1. Proof of Theorem 3.1

By definition of r,

$$\hat{Y}_v - Y_v = Y(\hat{\beta} - \beta^*) - r,$$

so

$$\|\hat{Y}_v - Y_v\|_2 \leq \|Y\|_2 \cdot \|\hat{\beta} - \beta^*\|_2 + \|r\|_2.$$

To bound $\|\hat{\beta} - \beta^*\|_2$, note that

$$(\hat{\tilde{\Sigma}} + \lambda I)\hat{\beta} = \hat{\tilde{\gamma}}, \quad (\tilde{\Sigma} + \lambda I)\beta^* = \tilde{\gamma}.$$

Subtracting the two equations gives

$$(\hat{\tilde{\Sigma}} + \lambda I)(\hat{\beta} - \beta^*) = (\tilde{\Sigma} - \hat{\tilde{\Sigma}})\beta^* + (\hat{\tilde{\gamma}} - \tilde{\gamma}).$$

Left-multiplying both sides by $(\hat{\tilde{\Sigma}} + \lambda I)^{-1}$ and taking norms gives

$$\|\hat{\beta} - \beta^*\|_2 \leq \frac{1}{\sigma_{\min}(\hat{\tilde{\Sigma}}) + \lambda} \left(\|\hat{\tilde{\Sigma}} - \tilde{\Sigma}\|_2 \cdot \|\beta^*\|_2 + \|\hat{\tilde{\gamma}} - \tilde{\gamma}\|_2 \right),$$

where we used the fact that $\|(\hat{\tilde{\Sigma}} + \lambda I)^{-1}\|_2^{-1} = \sigma_{\min}(\hat{\tilde{\Sigma}}) + \lambda$. This yields the first part of the theorem.

To bound $\|r\|_2$, note that

$$\begin{aligned} Y_v - Yb &= Uv + \mathcal{E}_v - UVb - \mathcal{E}b \\ &= Ue + \mathcal{E}_v - \mathcal{E}b. \end{aligned}$$

So,

$$\begin{aligned} r = Y_v - Y\beta^* &= Y_v - Yb + Yb - Y\beta^* \\ &= Ue + (\mathcal{E}_v - \mathcal{E}b) + Y(b - \beta^*), \end{aligned}$$

which implies

$$\|r\|_2 \leq \|Ue\|_2 + \|\mathcal{E}_v - \mathcal{E}b\|_2 + \|Y\|_2 \cdot \|b - \beta^*\|_2.$$

To bound $\|b - \beta^*\|_2$, we subtract $(\tilde{\Sigma} + \lambda I)b$ from both sides of $(\tilde{\Sigma} + \lambda I)\beta^* = \tilde{\gamma}$ to get

$$(\tilde{\Sigma} + \lambda I)(\beta^* - b) = (\tilde{\gamma} - \tilde{\Sigma}b) - \lambda b.$$

Left-multiplying both sides by $(\tilde{\Sigma} + \lambda I)^{-1}$ and taking norms gives

$$\|\beta^* - b\|_2 \leq \frac{1}{\sigma_{\min}(\tilde{\Sigma}) + \lambda} \left(\|\tilde{\gamma} - \tilde{\Sigma}b\|_2 + \lambda \|b\|_2 \right),$$

where we used the fact that $\|(\tilde{\Sigma} + \lambda I)^{-1}\|_2^{-1} = \sigma_{\min}(\tilde{\Sigma}) + \lambda$. This yields the second part of the theorem.

G.2. Proof of Theorem 5.1

By Assumptions E.1, E.2, and E.3, for each $i \in [n]$ and $k \in [K]$,

$$\begin{aligned} \mathbb{P}(\tilde{Y}_{i,m+1} = k \mid \tilde{u}_i) &= \langle \tilde{u}_{i,k}, v_{m+1,k} \rangle \\ &= \sum_{j=1}^{m} c_j \langle \tilde{u}_{i,k}, v_{j,k} \rangle \\ &= \sum_{j=1}^{m} c_j \mathbb{P}(\tilde{Y}_{i,j} = k \mid \tilde{u}_i), \end{aligned}$$

so $\tilde{P}_{m+1}(\cdot \mid \tilde{u}_i) = \sum_{j=1}^m c_j \tilde{P}_j(\cdot \mid \tilde{u}_i)$. Taking expectation on both sides yields $\mathbb{P}(\tilde{Y}_{i,m+1} = k) = \sum_{j=1}^m c_j \mathbb{P}(\tilde{Y}_{i,j} = k)$, so $\tilde{P}_{m+1} = \sum_{j=1}^m c_j \tilde{P}_j$. Similarly, $P_{m+1} = \sum_{j=1}^m c_j P_j$. Thus, denoting $\hat{P}_j = \sum_{i=1}^n \hat{w}_i \tilde{P}_j(\cdot \mid \tilde{u}_i) + \sum_{k=1}^K \hat{\pi}_k \delta_k$ and in particular $\hat{P}_{m+1} = \sum_{i=1}^n \hat{w}_i \tilde{P}_{m+1}(\cdot \mid \tilde{u}_i) + \sum_{k=1}^K \hat{\pi}_k \delta_k$, we have

$$
\begin{aligned}
&\mathrm{TV}\left(P_{m+1},\ \hat{P}_{m+1}\right) \\
&= \mathrm{TV}\left(\sum_{j=1}^m c_j P_j,\ \sum_{j=1}^m c_j \hat{P}_j\right) \\
&\leq \sum_{j=1}^m c_j \mathrm{TV}\left(P_j,\ \hat{P}_j\right) \\
&\leq m\|c\|_\infty \cdot \frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \hat{P}_j\right) \\
&= m\|c\|_\infty \cdot \min_{(w,\pi)\in\Delta^{n+K-1}} \left\{\frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \hat{P}_j(w,\pi)\right)\right\}, \qquad (9)
\end{aligned}
$$

where $\hat{P}_j(w,\pi) = \sum_{i=1}^n w_i \tilde{P}_j(\cdot \mid \tilde{u}_i) + \sum_{k=1}^K \pi_k \delta_k$. Let ν^* denote the optimal reweighting:

$$
\nu^* = \mathrm{argmin}_\nu \frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \sum_{u\in\mathcal{S}} \nu(u)\tilde{P}_j(\cdot \mid u)\right), \qquad (10)
$$

where the argmin is taken over all reweightings $\nu : \mathcal{S} \to [0,1]$ satisfying $\sum_{u\in\mathcal{S}} \nu(u) = 1$ and $\nu(u) \leq A\tilde{\mu}(u)$ for all $u \in \mathcal{S}$. It induces a reweighting on the empirical distribution formed by the n digital twins:

$$
w_i^* = \frac{1}{Z}\cdot\frac{1}{n}\cdot\frac{\nu^*(\tilde{u}_i)}{\tilde{\mu}(\tilde{u}_i)} \quad \forall i \in [n],
$$

where

$$
Z = \frac{1}{n}\sum_{i=1}^n \frac{\nu^*(\tilde{u}_i)}{\tilde{\mu}(\tilde{u}_i)}.
$$

(We follow the convention that $0/0 = 0$.) Clearly, $w_i^* \geq 0$ for all $i \in [n]$, and $\sum_{i=1}^n w_i^* = 1$. Then, we can bound (9) via

$$
\begin{aligned}
&\min_{(w,\pi)\in\Delta^{n+K-1}} \left\{\frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \hat{P}_j(w,\pi)\right)\right\} \\
&\leq \frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \sum_{i=1}^n w_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right) \\
&\leq \frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(P_j,\ \sum_{u\in\mathcal{S}} \nu^*(u)\tilde{P}_j(\cdot \mid u)\right) \qquad (11) \\
&+ \frac{1}{m}\sum_{j=1}^m \mathrm{TV}\left(\sum_{u\in\mathcal{S}} \nu^*(u)\tilde{P}_j(\cdot \mid u), \sum_{i=1}^n w_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right). \qquad (12)
\end{aligned}
$$

The term (11) is simply (10). We now work on the term (12). Let $r_i^* = Zw_i^* = n^{-1} \cdot \nu^*(\tilde{u}_i)/\tilde{\mu}(\tilde{u}_i)$, then for each $j \in [m]$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right] \\
&= \sum_{i=1}^{n} \sum_{u \in \mathcal{S}} \tilde{\mu}(u) \cdot \left(\frac{1}{n} \cdot \frac{\nu^*(u)}{\tilde{\mu}(u)} \cdot \tilde{P}_j(\cdot \mid u)\right) \\
&= \sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u).
\end{aligned}$$

Therefore, using the fact that $\mathrm{TV}(P, Q) = \frac{1}{2}\|P - Q\|_1$ for all probability distributions P and Q, we have

$$\begin{aligned}
&\frac{1}{m} \sum_{j=1}^{m} \mathrm{TV}\left(\sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u),\ \sum_{i=1}^{n} w_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right) \\
&= \frac{1}{2m} \sum_{j=1}^{m} \left\| \sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u) - \sum_{i=1}^{n} w_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) \right\|_1 \\
&= \frac{1}{2Zm} \sum_{j=1}^{m} \left\| \sum_{u \in \mathcal{S}} Z\nu^*(u) \tilde{P}_j(\cdot \mid u) - \sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) \right\|_1 \\
&\leq \frac{1}{2Zm} \sum_{j=1}^{m} \left(|1 - Z| \left\| \sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u) \right\|_1 + \left\| \sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u) - \sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) \right\|_1 \right) \\
&= \frac{1}{2Z} \left(|1 - Z| + \frac{1}{m} \sum_{j=1}^{m} \left\| \sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right] \right\|_1 \right), \qquad (13)
\end{aligned}$$

where the inequality follows from the triangle inequality.

We will now apply Hoeffding's inequality to obtain high-probability bounds on $|1 - Z|$ and Z. Clearly, $\{\nu^*(\tilde{u}_i)/\tilde{\mu}(\tilde{u}_i)\}_{i=1}^n$ are i.i.d. By definition, $0 \leq \nu^*(\tilde{u}_i)/\tilde{\mu}(\tilde{u}_i) \leq A$ for all $i \in [n]$. Moreover, since $\tilde{u}_i \sim \tilde{\mu}$, then

$$\mathbb{E}\left[\frac{\nu^*(\tilde{u}_i)}{\tilde{\mu}(\tilde{u}_i)}\right] = \sum_{s \in \mathcal{S}} \frac{\nu^*(s)}{\tilde{\mu}(s)} \cdot \tilde{\mu}(s) = \sum_{s \in \mathcal{S}} \nu^*(s) = 1.$$

Note that the event

$$\left\{ |Z - 1| \leq A\sqrt{\frac{\log(4/\alpha)}{2n}} \right\}$$

expands to

$$\left\{ \left| \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\nu^*(\tilde{u}_i)}{\tilde{\mu}(\tilde{u}_i)} - \mathbb{E}\left[\frac{\nu^*(\tilde{u}_i)}{\tilde{\mu}(\tilde{u}_i)}\right] \right) \right| \leq A\sqrt{\frac{\log(4/\alpha)}{2n}} \right\}.$$

By Hoeffding's inequality,

$$\mathbb{P}\left(|Z - 1| \leq A\sqrt{\frac{\log(4/\alpha)}{2n}} \right) \geq 1 - \frac{\alpha}{2}.$$

When this event happens, if $n \geq 2A^2 \log(4/\alpha)$, then $|Z - 1| \leq 1/2$, which implies $Z \geq 1/2$. Substituting this into (13) yields that with probability at least $1 - \alpha/2$,

$$\begin{aligned}
&\frac{1}{m} \sum_{j=1}^{m} \mathrm{TV}\left(\sum_{u \in \mathcal{S}} \nu^*(u) \tilde{P}_j(\cdot \mid u),\ \sum_{i=1}^{n} w_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right) \\
&\leq A\sqrt{\frac{\log(4/\alpha)}{2n}} + \frac{1}{m} \sum_{j=1}^{m} \left\| \sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right] \right\|_1. \qquad (14)
\end{aligned}$$

It remains to bound the second term on the right-hand side of (14). We will use McDiarmid's inequality (e.g., Theorem 2.9.1 in Vershynin (2018)). To this end, we view it as a function of $(\tilde{u}_1, ..., \tilde{u}_n)$. Define

$$\varphi(u'_1, ..., u'_n) = \frac{1}{m}\sum_{j=1}^{m}\left\|\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid u'_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right]\right\|_1.$$

For all $\ell \in [n]$ and for all $(u'_1, ..., u'_n)$ and $(u''_1, ..., u''_n)$ with $u'_j = u''_j \ \forall j \neq \ell$, by the triangle inequality,

$$\begin{aligned}
&\left|\varphi(u'_1, ..., u'_n) - \varphi(u''_1, ..., u''_n)\right| \\
&\leq \frac{1}{m}\sum_{j=1}^{m}\left\|\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid u'_i) - \sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid u''_i)\right\|_1 \\
&\leq \frac{1}{m}\sum_{j=1}^{m} r_\ell^* \left\|\tilde{P}_j(\cdot \mid u'_\ell) - \tilde{P}_j(\cdot \mid u''_\ell)\right\|_1 \\
&\leq 2r_\ell^* = \frac{2}{n}\cdot\frac{\nu^*(\tilde{u}_\ell)}{\tilde{\mu}(\tilde{u}_\ell)} \leq \frac{2A}{n}.
\end{aligned}$$

By McDiarmid's inequality (e.g., Theorem 2.9.1 in Vershynin (2018)), with probability at least $1 - \alpha/2$,

$$\begin{aligned}
&\frac{1}{m}\sum_{j=1}^{m}\left\|\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right]\right\|_1 \\
&\leq A\sqrt{\frac{\log(2/\alpha)}{2n}} + \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}\left\|\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right]\right\|_1. \qquad (15)
\end{aligned}$$

To bound the expectation in (15), for each $j \in [m]$,

$$\begin{aligned}
&\mathbb{E}\left\|\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(\cdot \mid \tilde{u}_i)\right]\right\|_1 \\
&= \sum_{k=1}^{K}\mathbb{E}\left|\sum_{i=1}^{n} r_i^* \tilde{P}_j(k \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(k \mid \tilde{u}_i)\right]\right| \\
&\leq \left[K\sum_{k=1}^{K}\left(\mathbb{E}\left|\sum_{i=1}^{n} r_i^* \tilde{P}_j(k \mid \tilde{u}_i) - \mathbb{E}\left[\sum_{i=1}^{n} r_i^* \tilde{P}_j(k \mid \tilde{u}_i)\right]\right|\right)^2\right]^{1/2} && \text{(Cauchy-Schwarz)} \\
&\leq \sqrt{K\sum_{k=1}^{K}\mathrm{var}\left(\sum_{i=1}^{n} r_i^* \tilde{P}_j(k \mid \tilde{u}_i)\right)} \\
&= \sqrt{K\sum_{k=1}^{K}\sum_{i=1}^{n}(r_i^*)^2\mathrm{var}\big(\tilde{P}_j(k \mid \tilde{u}_i)\big)} && \text{(independence)} \\
&\leq \sqrt{K\sum_{k=1}^{K}\sum_{i=1}^{n}\left(\frac{A}{n}\right)^2 \tilde{P}_j(k)} \\
&\leq A\sqrt{\frac{K}{n}}, \qquad (16)
\end{aligned}$$

where the second last inequality uses the fact that for a random variable $X \in [0, 1]$, it holds that $\mathrm{var}(X) \leq \mathbb{E}[X^2] \leq \mathbb{E}[X]$.

Substituting (12), (14) and (16) into (9), and using a union bound, we obtain that with probability at least $1-\alpha$,

$$\begin{aligned}
&\mathrm{TV}\left(P_{m+1},\ \sum_{i=1}^{n}\hat{w}_i\tilde{P}_{m+1}(\cdot \mid \tilde{u}_i)+\sum_{k=1}^{K}\hat{\pi}_k\delta_k\right)\\
&\leq m\|c\|_\infty\cdot\left[\inf_{\nu}\frac{1}{m}\sum_{j=1}^{m}\mathrm{TV}\left(P_j,\ \sum_{u\in\mathcal{S}}\nu(u)\tilde{P}_j(\cdot \mid u)\right)+A\sqrt{\frac{\log(4/\alpha)}{2n}}+A\sqrt{\frac{K}{n}}+A\sqrt{\frac{\log(2/\alpha)}{2n}}\right]\\
&\leq m\|c\|_\infty\cdot\left[\inf_{\nu}\frac{1}{m}\sum_{j=1}^{m}\mathrm{TV}\left(P_j,\ \sum_{u\in\mathcal{S}}\nu(u)\tilde{P}_j(\cdot \mid u)\right)+\sqrt{3}A\sqrt{\frac{K+\log(4/\alpha)}{n}}\right].
\end{aligned}$$

This finishes the proof.