

ExploraTwin, a Non-Profit Research Platform for Digital Twin Simulations

Naveen Venkatanarayanan

Columbia University
nv2444@columbia.edu

Yuchen Qiu

Columbia Business School
yq2411@columbia.edu

Tianyi Peng

Columbia Business School
tianyi.peng@columbia.edu

George Gui

Columbia Business School
zg2467@gsb.columbia.edu

Olivier Toubia

Columbia Business School
ot2107@gsb.columbia.edu

ABSTRACT

Digital twin simulations show promise, but current empirical evidence suggests that the approach should be tested before being deployed in any particular context. To lower the friction for researchers and practitioners to test and deploy digital twin simulations, this brief commentary introduces ExploraTwin, an open-access, non-profit research platform for digital twin survey simulations. ExploraTwin supports two modes. In survey mode, researchers can upload a Qualtrics survey file or create a survey within the platform; select an available sample of digital twins; configure and run the simulation, and export analysis-ready data. In panel mode, researchers can assemble a small group of twins for open-ended conversations, document annotation, and moderated, focus-group-style voice discussions. We also developed CroissantTwin, a standardized data format for adding samples of digital twins to the platform. We demonstrate the survey mode workflow by using the platform to replicate 19 experiments on digital twins from the Twin-2K-500 dataset. ExploraTwin’s survey execution fidelity is high: 99.6% of 197,000 answer units returned a structurally valid response on the first run.

Keywords: digital twins, large language models (LLMs), consumer research, experiment simulation, synthetic personas.

Large language models (LLMs) offer researchers the *potential* to field a survey to an entire panel of simulated respondents almost instantaneously, at a small fraction of the cost of human data collection, using instruments and stimuli that have never been shown to anyone (Aher, Arriaga, and Kalai 2023; Argyle et al. 2023; Ashokkumar et al. 2026; Manning and Horton 2026; Park et al. 2023; Peng et al. 2025; Toubia et al. 2025).

A particularly promising approach for doing so leverages digital twins, i.e., prompting LLMs with persona information—individuals’ attributes, behavioral histories, survey responses, or open-ended reflections—to simulate specific human respondents in experiments (Park et al. 2024; Peng et al. 2025; Toubia et al. 2025). For academic researchers and market research practitioners, digital twins potentially offer a fast and economical way to pretest survey materials, explore hypotheses, screen market research questions before costly human data collection, or revisit completed studies with additional questions.

So far, the predictive performance of digital twins has been mixed (Park et al. 2024; Peng et al. 2025; Toubia et al. 2025). This underscores the need for researchers and practitioners to experiment with and test digital twin pipelines before deploying them for a particular use case. This in turn requires tools that enable researchers to run studies on digital twins at low cost and with low friction. Despite the low marginal API cost per synthetic respondent, current solutions typically involve either subscribing to a commercial tool or developing one’s own digital twin pipeline. The latter presents at least two types of friction.

First, simulation pipelines are costly to engineer because consumer research uses a wide range of instruments and potentially complex structures. Consider a Qualtrics survey as an example. It might contain a mix of open-ended, scale, multiple choice questions, as well as features like block randomization, branching, loop and merge, etc. Simulating such a study requires tools that can parse the instrument, execute or flag its logic, and return analysis-ready data. A shared workflow for adapting diverse study formats would substantially reduce the engineering burden on researchers.

Second, samples of digital twins, which we refer to as “persona banks,” are difficult to

transport across studies and tools. A persona bank is a dataset containing a sample of synthetic personas. Over the last two years, several such datasets have been created. These datasets differ in population, structure, and form of the persona itself: prior survey answers with structured and unstructured attributes (Toubia et al. 2025), interview-derived scripts (Park et al. 2024), behavioral logs (Wang et al. 2026), and synthetic profile collections and narratives (Ge et al. 2024; NVIDIA Corporation 2026). Without a common standard that tells the simulation tool how to understand them, researchers cannot easily use different persona datasets in the same simulation workflow. As a result, each new persona dataset requires a custom way to load and process it for use in a simulation.

In this brief commentary, we introduce ExploraTwin (<https://exploratwin.org>), an open-access, non-profit research platform that implements a workflow for running digital twin survey simulations with minimal cost and friction. In terms of cost, ExploraTwin is currently free (with a monthly allowance of \$10 in usage credits). Future versions might charge nominally to cover API and development costs (on the order of \$0.01 per synthetic respondent). In terms of friction, the platform is able to handle a wide range of instruments, allowing users, for example, to simply upload a Qualtrics survey file (.qsf) and receive simulated data in the same format as with human respondents (.csv), in a matter of minutes. While the platform’s primary persona bank is the Twin-2K-500 sample of digital twins (Toubia et al. 2025), users can also choose from various other persona banks. CroissantTwin further provides a standardized data format for creating new persona banks (e.g., focused on specific regions or topics).

In sum, ExploraTwin turns digital twin simulation from a heavy engineering problem into a standardized workflow that social science researchers and market research practitioners can inspect, reuse, and evaluate. We release an open-source package for ExploraTwin’s survey-mode pipeline on GitHub (<https://github.com/nav-v/surveytwin-oss.git>) and also maintain a hosted non-profit web instance (<https://exploratwin.org>).

The rest of this brief commentary proceeds as follows. First, we introduce the ExploraTwin workflow and walk through the two simulation modes (survey and panel). Second, we

describe CroissantTwin, our protocol for adding new persona banks to the platform. Third, we examine survey fidelity by replicating the 19 Mega-study experiments conducted by Peng et al. (2025). Finally, we discuss cost efficiency, limitations, and directions for future research.

THE EXPLORATWIN PIPELINE: SURVEY MODE

ExploraTwin offers two ways to conduct studies with digital twins, distinguished by the form of response the researcher wants to produce. Survey mode works like fielding an online survey to a sample of respondents. Researchers can either (i) upload a Qualtrics survey file (.qsf), or (ii) compose a survey in an interactive builder, either starting from a blank canvas with drag-and-drop editing or from a draft that ExploraTwin generates from a typed research question. The panel mode, introduced in the next section, allows the user to assemble a small group of twins in a conversational interface for qualitative interviews, document annotation, and focus-group-style discussions.

Figure 1 summarizes the survey-mode workflow from researcher input through prompt construction, configuration, response generation, validation, repair, and export. Execution depends on the survey’s logic. In static surveys, each twin completes the full survey in one model call: the questionnaire is presented once, and the model returns all answers in the requested format. If later questions depend on earlier answers (e.g., branching based on an earlier response), the run is staged: ExploraTwin first collects the answers needed to resolve the next part of the survey, applies the flow logic, and then presents the remaining questions with the twin’s earlier answers.

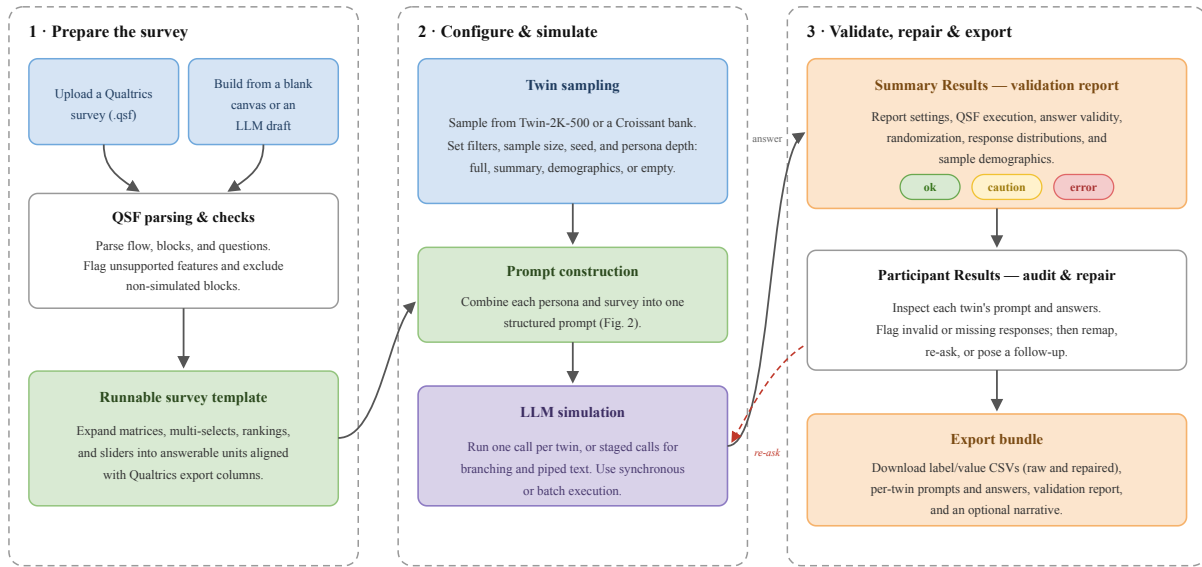


FIGURE 1: SURVEY-MODE WORKFLOW FROM PREPARATION THROUGH VALIDATION AND EXPORT

Preparing the Survey

ExploraTwin provides two ways to prepare a survey to be fielded to the selected twins.

From a QSF. The researcher uploads a survey in Qualtrics’ native format, .qsf. The ExploraTwin parser then automatically converts the file into a runnable survey template. The parser supports a broad set of Qualtrics question types and logic features. See a complete list in table A2. If the uploaded survey contains features the pipeline cannot fully reproduce, such as custom Qualtrics JavaScript or externally defined embedded-data fields, ExploraTwin surfaces warnings before the run rather than silently translating them. The researcher can inspect the parsed block structure and drop blocks that should not be simulated, such as consent, debriefing, or administrative blocks.

The parser also supports the inclusion of images in the survey. If the QSF contains visual

stimuli, ExploraTwin automatically extracts them and pairs them with the surrounding text so vision-capable AI models can consider both when responding.

From the Survey Builder. Alternatively, the researcher can compose the survey in an interactive builder, starting either from a blank canvas or from a generated draft. In the latter case, the researcher types a research question or brief, and an LLM agent prompted as a survey designer drafts a structured questionnaire as a starting point. In both cases, the survey can include single-choice items, multi-select questions, rating scales, matrix items, open text, and bounded numeric-entry questions. And the researcher revises it in place by editing question wording, adding or removing items, attaching image stimuli, or requesting natural-language edits from the survey-design assistant. Once approved, the survey is converted into the same parsed-template structure used by the QSF pipeline. The builder targets straightforward linear instruments; complex survey logic—randomization, branching, display logic, piped text—requires the QSF path described above.

From Survey to Simulation Prompt

An effective simulation is not just pasting survey questions into a prompt. Human respondents experience a survey as an interactive instrument, with the interface controlling question order, branching, display rules, and allowed response formats. By contrast, an LLM reads the survey as a token sequence. Therefore, the survey has to be translated into an LLM-readable template that preserves this survey-taking logic while specifying how the model should view and answer each question. For example, a human respondent answers a slider by dragging a control between labeled endpoints, whereas the LLM receives the question and endpoint labels as text, together with the permitted numeric range and an instruction to return a single value in the required format. The description of the selected persona is then combined with this template, and the model is instructed to return choices or text in a structured format. This translation layer lets

the simulation approximate the logic of human survey-taking while working within the input and output constraints of LLMs.

For each twin, the full simulation prompt combines four components (see figure 2). A system instruction tells the model to answer as a digital twin of a human and defines the behavioral rules for the run. A persona representation describes the respondent being simulated. An LLM-readable survey component presents the question text, response options, stimuli, validation rules, and relevant flow context in a form the model can process. Finally, an output schema specifies how the model should return a parseable answer.

We note a distinction between survey questions and answer units. A survey question is the item displayed to a respondent, whereas an answer unit is the smallest response field expected by the instrument and recorded as a separate column in the exported data. Simple questions, such as single-choice or open-text items, generally produce one answer unit. Complex questions produce several: a matrix produces one unit for each row or cell; a multi-select question produces one selection decision for each option; a rank-order question produces one rank for each alternative; and a multi-statement slider produces one numeric response for each statement. The LLM is instructed to return a separate structured response for every answer unit presented to it. This does not mean that each unit requires a separate model call; one call may contain many questions and answer units.

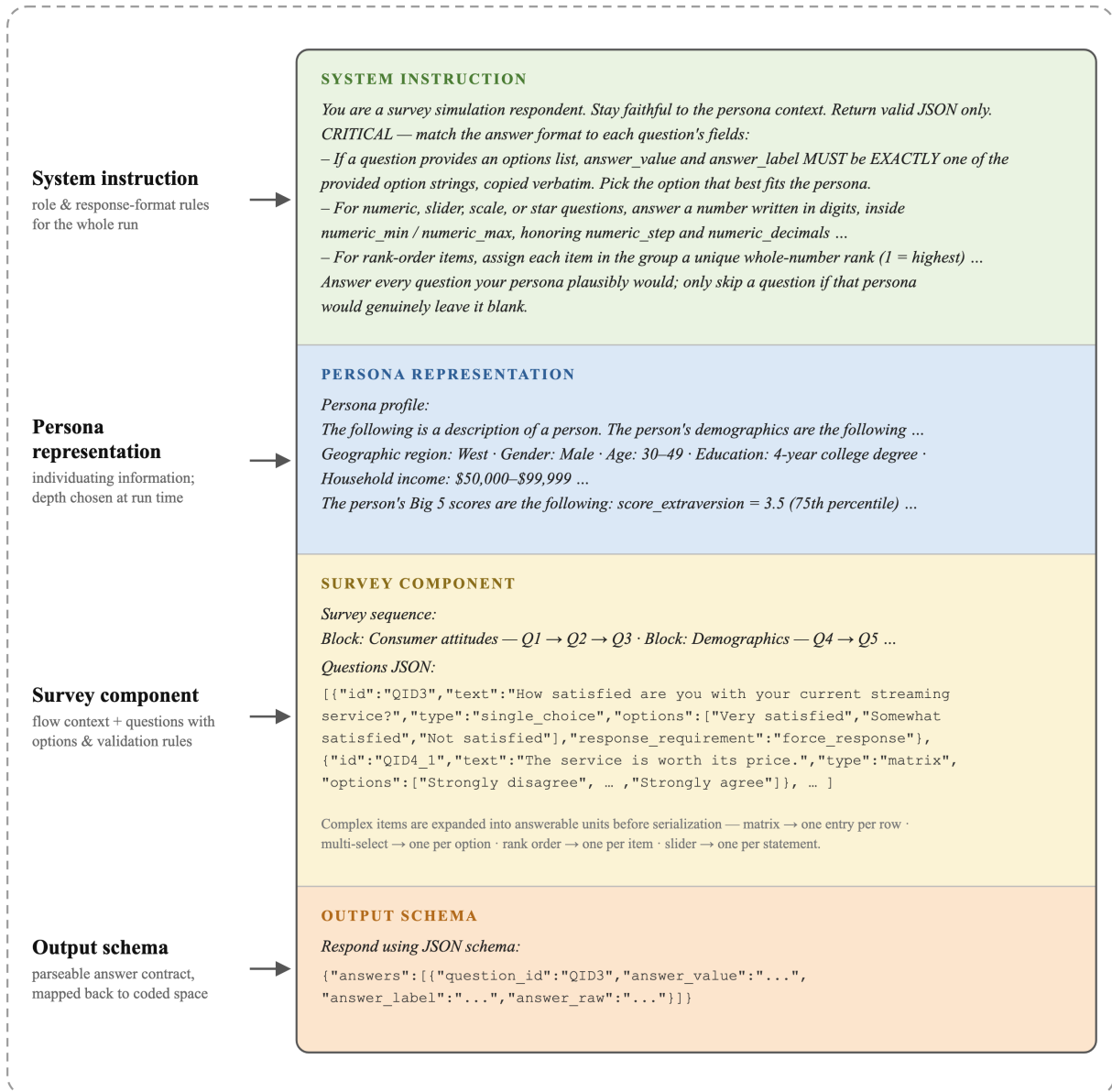


FIGURE 2: SIMULATION PROMPT STRUCTURE

Configuring the Run

Once the survey is prepared, the researcher configures who answers it and how the simulation is executed. Figure 3 shows the configuration page, where researchers choose the persona bank, population filters, model, and execution settings.

- **Model.** The researcher chooses the model (LLM) used for the simulation.
- **Persona bank.** The researcher chooses which persona bank to field the survey on: the default Twin-2K-500 bank, another bank available from the platform, or a custom bank uploaded by the user at run time (see the CroissantTwin section below). The chosen bank determines which representations and filters (e.g., demographic characteristics) are available.
- **Persona representation kind.** The researcher chooses what kind of persona information to supply to the LLM. For each respondent, the selected kind determines which representation is used. For instance, the built-in Twin-2K-500 bank offers three representation kinds: *full* (every question–answer pair from the respondent’s original 500-question record), *summary* (a paragraph-length narrative distilled from that record), and *demographics-only* (demographic attributes alone). Researchers may also select an *empty* baseline, which includes no respondent-specific information, so responses reflect only the base LLM.
- **Sample size.** The researcher chooses the number of twins to simulate.
- **Population filters.** The researcher can narrow the eligible pool using the filterable attributes associated with that bank. In the built-in Twin-2K-500 bank, available filters include region, age band, sex at birth, income bracket, education, and other demographic variables.
- **Report options.** The researcher can choose whether to receive only the default exports and deterministic summary-results report, or to request an optional AI-written narrative report.
- **Calling method.** The researcher can choose either synchronous calls or batch calling. Batch calling takes longer to complete but reduces token cost by roughly half.

The screenshot displays the ExplorATwin configuration interface, organized into three main sections:

- UPLOADED SURVEY:** A green banner indicates a "SURVEY LOADED" titled "Accuracy Nudge Pennycook et al. 2021 Study 5_Replication" with 34 blocks. A note below states: "Supports standard block parsing, question and answer-choice images stored in the QSF, Qualtrics randomizers, display and branch logic, QSF-defined embedded data, and static Loop 5 Merge tables. Does not support JavaScript-driven survey behavior or randomization, or flows that depend on external CSV or web-service data."
- MODEL & EXECUTION:** This section contains several controls:
 - PERSONA BANK:** A dropdown menu set to "Twin-2K-500 (built-in)" with buttons for "Browse banks" and "+ Create your own".
 - MODEL:** A dropdown menu set to "GPT-4o Mini".
 - PERSONA:** A dropdown menu set to "Summary" with a description: "Lower cost, faster, balanced fidelity."
 - AI REPORT:** A toggle switch labeled "EXPERIMENTAL" with the instruction: "Generate an AI-written narrative report after the simulation."
 - SAMPLE SIZE:** A dropdown menu set to "40".
 - SEED (optional):** A text input field containing "e.g. 42" with a note: "Reproducible sample selection."
- AUDIENCE FILTERS:** This section shows a "Total audience: 2,058 twins" and various filter categories:
 - REGION:** Northeast, Midwest, South, West.
 - AGE BAND:** 18-29, 30-49, 50-64, 65+.
 - SEX AT BIRTH:** Male, Female.
 - INCOME BRACKET:** Less than \$30,000, \$30,000-\$50,000.
 - EDUCATION:** Less than high school, High school graduate.
 - RACE ORIGIN:** White, Black, Asian, Hispanic, Other.

FIGURE 3: CONFIGURATION PAGE OVERVIEW ON EXPLORATWIN

Before running the survey, the website provides two previews. First, the survey preview lets the researcher inspect the parsed survey and the included blocks. Second, the prompt preview shows a sample of full simulation prompts. The researcher can edit the system prompt at this stage to add simulation-specific instructions, while the platform records the final prompt setting as part of the run.

Post-Simulation Validation

When the simulation finishes, the pipeline provides diagnostics and descriptive summaries of the results. A digital twin survey run raises two practical questions: whether the survey was administered as intended and whether the returned answers are structurally usable. ExplorATwin addresses these questions through two linked result views: *Summary Results* and *Participant Results*.

Summary Results. The Summary Results portal is an interactive dashboard that brings together diagnostics of configuration, execution, and responses before the data are used for analysis. Diagnostics are computed from the parsed survey, run configuration, randomization assignments, and returned responses. The Summary Results view has six main components: experiment setting, QSF fidelity, response validity, randomization and balance, descriptive statistics, and persona coverage, with an overall status summary to guide review. Table A1 in the appendix summarizes the information reported for each component. Its purpose is to show whether the survey was administered as intended, whether the returned answers satisfy the survey’s structural requirements, and which responses require review.

Participant Results. The same results interface also provides a participant-level view in which researchers can inspect each twin’s prompt and answers, filter for flagged responses, and repair selected abnormal answers. Even when a survey has been converted into an LLM-readable format, a small share of responses can still be incomplete or format-invalid. For example, a twin may skip a forced-response question, answer outside the provided option set, or return a response that violates the requested output template. The participant view flags three main types of problematic responses: answers that violate the output template, answers outside the provided options or allowed range, and skipped forced-response questions. ExploraTwin provides two repair options, rematching and rerunning, to resolve these errors. Web appendix A describes the rematching rules and rerun procedure in detail. Researchers can also ask a specific twin follow-up questions using the chat box below its responses. This individual view supports auditing and interpretation.

Validation Checks on Simulations. How can researchers determine whether a simulation reproduced the study they intended to field? Our guiding principle is that the pipeline should preserve the survey’s intended exposure and response structure, or clearly flag where it cannot. We therefore include in the appendix a table that lists question formats, question settings, and survey

flow and logic that are supported, partially supported, or unsupported. See table A2.

Retrieving Outputs

Every survey-mode run produces a downloadable bundle. First, the bundle includes CSV response files in a Qualtrics-like format, with answers stored both as text labels and as coded values, and with separate files for raw and repaired data. Second, it includes per-twin prompt and answer files: one question file and one answer file for each twin, numbered by twin ID. Third, it includes the deterministic validation report, which records the Summary Results diagnostics and a compact run-level status of *ok*, *caution*, or *error* based on problematic questions and failed calls (see table A1 for details). The status is a guide to review; the underlying flags remain visible. A simulation with a *caution* status may still be usable after review, but it may require rematching answers the twins already gave or rerunning missing and out-of-range answers. Conversely, a simulation with apparently plausible response distributions may be unusable if the validation report shows that a key branch, stimulus, or response constraint was not faithfully administered. Optionally, ExploraTwin can also produce an AI-generated analysis report that is separate from the deterministic validation report. Its numerical inputs come from precomputed statistics, while the LLM layer writes narrative summaries based on those statistics.

Together, these files let researchers analyze the simulated data, inspect the prompts and answers behind each row, and share the run record with others.

THE EXPLORATWIN PIPELINE: PANEL MODE

Panel mode is a separate pipeline for qualitative research: the researcher assembles a small panel of digital twins and asks open-ended questions to elicit reactions, explanations, or critiques. Panel mode supports two forms of interaction: (i) a text-based workspace for panel

questions, direct follow-ups, and document review, and (ii) TwinMeet for moderated voice conversations.

Panel Configuration

Researchers begin by selecting a persona bank, a persona format, and an AI model. They can then filter candidate personas by specific criteria, such as demographic attributes in the built-in Twin-2K-500 dataset or custom traits defined in third-party CroissantTwin datasets. Matching AI personas appear in a preview list with avatars, screen names, and brief profiles. Researchers can select up to 10 personas to form a panel and adjust the roster at any point during the study.

Text-Based Conversations and Document Review

Panel Conversations. The researcher sends an open-ended question to the panel, and each twin's response appears in a separate participant card. Each twin answers from its selected persona representation and a private history containing the panel questions, its own direct questions, and its prior replies, but not other members' responses. Panel mode therefore functions as parallel interviews rather than a shared focus group, preventing one twin's response from directly anchoring another's. Researchers can also ask additional follow-up questions to specific twins in the thread.

Document Review. Researchers can upload images, PDFs, presentations, and text documents when they want the panel to review a stimulus rather than respond to a standalone prompt. The twins return annotations tied to specific passages, pages, cells, or visual locations. The review interface groups comments by location, allowing the researcher to compare how several twins responded to the same part of a document. Clicking an annotation initiates a direct follow-up exchange with the twin that produced it.

TwinMeet

TwinMeet provides a moderated live voice conversation with up to five twins. The interface resembles a videoconference-style layout, with the researcher and each twin represented by a participant tile. The researcher gives the session a title that establishes the conversational context for the twins, selects the participants, and grants the floor to one twin at a time. The selected twin responds by voice, while the remaining members follow a shared transcript and can signal when they want to contribute. Unlike text-based panel conversations, this shared context allows twins to react to, build on, or disagree with earlier remarks, making TwinMeet closer to a moderated focus group than to parallel interviews. Calls are limited to 10 minutes, and the transcript is saved in the panel thread.

Taken together, Panel Mode supports both independent qualitative elicitation through text and document review, as well as moderated group exchanges through TwinMeet. The researcher has access to the entire conversation and its associated materials for further analysis. Appendix figures A1, A2, and A3 illustrate the text-based conversation, document-review, and TwinMeet interfaces, respectively.

CROISSANTTWIN: A STANDARDIZED DATA FORMAT FOR PERSONA BANKS

By default, ExploraTwin uses Twin-2K-500 (Toubia et al. 2025), based on a dataset representative of the US population. However, researchers often need to study specific sub-segments (e.g., physicians, Gen Z gamers), international populations, or proprietary customer records. Furthermore, while Twin-2K-500 is a sample of *digital twins* that are each built to mimic one specific individual, researchers may also be interested in running simulations on other samples of synthetic respondents that are not necessarily tied to specific individuals, but rather to segments in the population. That is, a persona bank could be a sample of digital twins, or a sample of any

other type of synthetic personas. Consequently, as AI persona datasets proliferate, from PersonaHub’s billion-persona web text extractions (Ge et al. 2024) to NVIDIA’s global census-aligned Nemotron-Personas (NVIDIA Corporation 2026), the range of potential persona banks is expanding rapidly.

Currently, each dataset uses its own format for defining demographics, filtering criteria, and individual profiles. Integrating new persona banks currently requires writing custom adapters for every tool—a costly, repetitive process. To solve this, we introduced CroissantTwin, an open data standard that creates a unified format for packaging AI persona collections. Built as an extension of the MLCommons Croissant 1.1 dataset standard (Akhtar et al. 2024), CroissantTwin allows any CroissantTwin-formatted dataset to plug seamlessly into ExploraTwin (and other compatible platforms) without requiring custom code.

How a Persona Bank Works

A persona bank consists of a simple folder containing data files and a standardized manifest file. By tagging tables and columns with universal labels, the manifest allows platforms like ExploraTwin to load any CroissantTwin-formatted dataset automatically without custom code. Figure 4 illustrates the resulting data model.

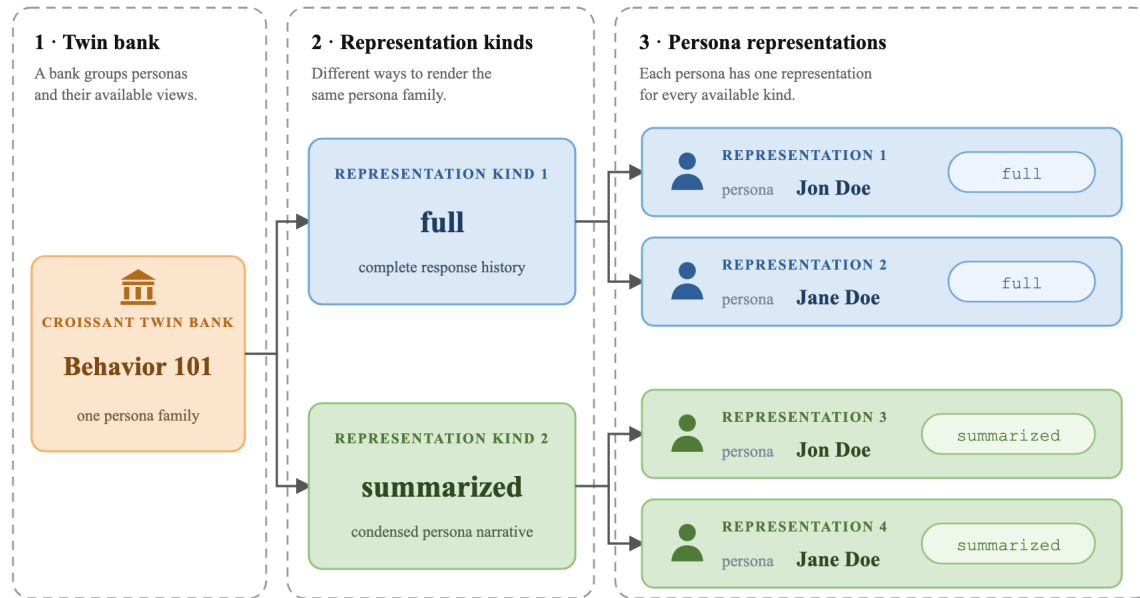


FIGURE 4: THE CROISSANTTWIN DATA MODEL: A BANK FANS OUT TO REPRESENTATION KINDS (THE SELECTABLE RENDERINGS); EACH KIND RESOLVES TO EXACTLY ONE REPRESENTATION PER PERSONA

The standard rests on three concepts. A *persona* is the fundamental unit of simulation. It represents an individual respondent, customer, or synthetic agent; it is identified by a unique ID and structured attributes, unstructured text, or both. A *representation* is a specific prompt formatted for the AI model. A persona can be rendered in multiple ways depending on research needs. *Kinds* are categories of representations defined by the persona bank creator. Each kind includes a clear usage note outlining its purpose, context, and limitations. When configuring a panel in ExploraTwin, researchers select a persona bank and then choose its representation kind. For instance, the built-in Twin-2K-500 dataset offers Full, Summary, and Demographics-only options, while third-party CroissantTwin banks define their own custom kinds.

The manifest also carries key metadata declarations to support responsible data use.

- **Filters:** The author marks which persona fields can serve as population filters and lists their values.
- **Persona basis:** A mandatory label declares how the personas were created. An Observed Individual corresponds to one real person; a Human Composite deliberately blends several

people or aggregate data; a Synthetic Persona is generated with no claimed correspondence to anyone; and a Mixed Basis bank combines these categories, with each persona row stating its own basis.

- **Cataloging and access:** In addition to standard license and description fields, each bank specifies its domain, geographic region, and access requirements to streamline platform discovery.

The Persona-Bank Library and Run-Time Uploads

Figure 5 shows how conformant banks are presented in the in-platform library.

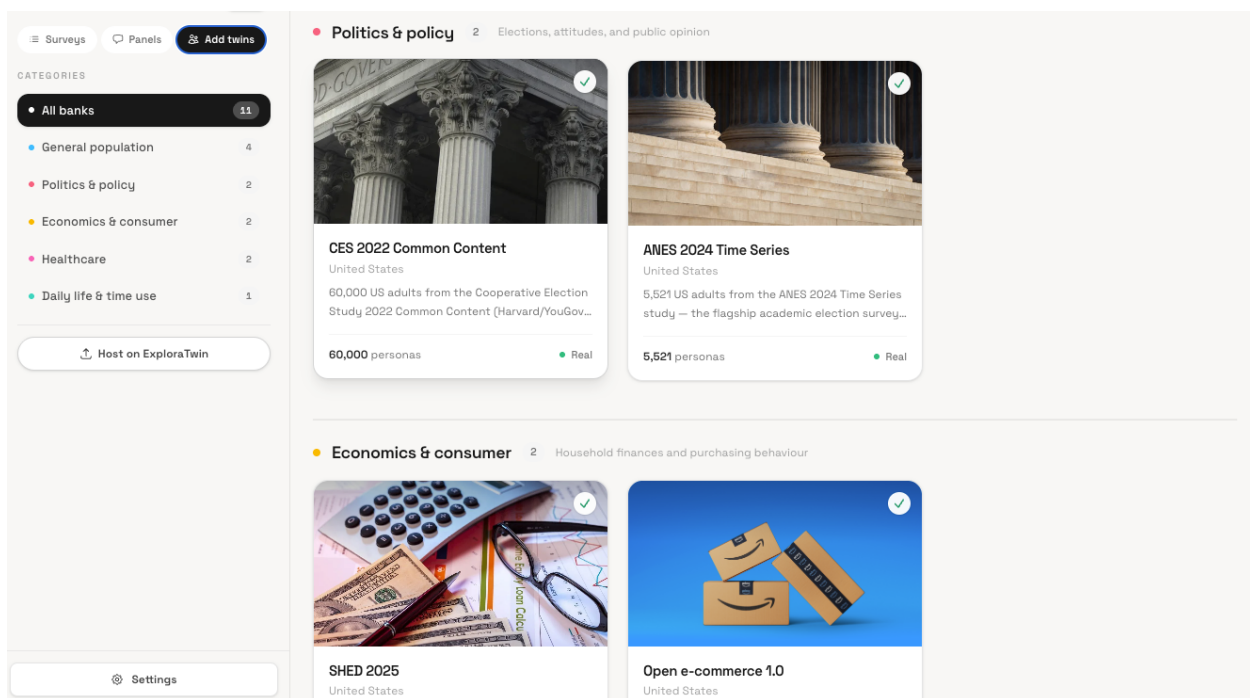


FIGURE 5: THE PERSONA-BANK LIBRARY

ExploraTwin supports two distinct operational modes for running persona banks:

- **Run-time uploads (private).** A bank can be supplied at run time, without being made available to other users. The user packages their data as a conformant bank and uploads it on the run-configuration page. The bank then appears in the researcher's own bank picker

with its available representations and filters, and can be used to execute studies seamlessly. These files remain private to the user’s active session, are never stored or published to the public library, and are automatically discarded after execution.

- **The persona-bank library (public).** The platform features an in-platform library that presents all hosted banks in a common catalog to be browsed, installed, and run. We populated it ourselves with 11 (at the time of writing) vetted banks chosen for range rather than size, including Twin-2K-500 (Toubia et al. 2025) and NVIDIA’s synthetic Nemotron narrative personas for Singapore (Thongpramoon et al. 2026). Importantly, we also include domain-specific banks derived from public survey microdata, such as the American National Election Studies’ 2024 Time Series Study (American National Election Studies 2026) in politics and policy and the National Health and Nutrition Examination Survey, August 2021–August 2023, in health (National Center for Health Statistics 2024). Researchers with datasets they would like published may reach out to the authors by clicking on the “Contribute Bank to ExploraTwin” button.

We provide two resources to lower the cost of building CroissantTwin banks: a public, versioned specification that defines the requirements for a conformant (i.e., properly formatted) bank, and a companion skill package for coding agents that turns any individual-level dataset into a validated, upload-ready bank (see web appendix B).

SURVEY-MODE FIDELITY: A 19-STUDY REPLICATION

We assess ExploraTwin’s survey mode by treating the original Qualtrics files from the 19 preregistered studies conducted by Peng et al. (2025) as new platform uploads. These studies span a broad range of topics and commonly used survey designs. Our primary objective is to evaluate execution fidelity: whether the pipeline administers each survey as intended, identifies structurally invalid responses, and records subsequent repairs.

Method

Digital-Twin Participants. We generated 5,700 study-level digital-twin response records to replicate the 19 experiments (Peng et al. 2025), with 300 respondents completing each study. We used the Twin-2K-500 dataset for persona profiles and selected its full persona representation, which preserves all 500 original questions spanning economics, psychology, and social science (Toubia et al. 2025). The original Twin-2K-500 dataset contains data from 2,058 human participants. We randomly selected 300 panel members and used the same 300 Full persona profiles across the 19 studies.¹ The twins were simulated with the gpt-5-mini base model with medium thinking effort.

Procedure. We replicated the 19 studies by using their original Qualtrics files (.qsf), running each study through the ExploraTwin survey-mode workflow. We used batch calling for static surveys and synchronous calling for surveys in which some questions depended on previous answers. Each study produced an export bundle identical to what a user would see on the website.

Cost. A central appeal of digital-twin simulation is that it changes the economics of early-stage experimentation. The first-pass simulations cost \$58.55 in API calls for 5,700 completed twins and 197,000 question-answer units; targeted repair added \$1.94, for a total of \$60.49. This is about 1.1 cents per completed twin, or about 0.9 cents per model call, even under the most information-rich persona setting. The per-respondent cost is therefore far below typical paid human-panel costs, making digital-twin panels especially attractive for screening ideas, testing survey wording, and comparing alternative stimuli before fielding a human study.

To examine how simulation costs vary with model choice and survey length, we selected five of the 19 experiments, spanning surveys with 2 to 63 questions, and reran each using three models currently available in ExploraTwin: gpt-4o-mini, gpt-5.6 Luna, and gpt-5-mini. All

¹Using the same set of persona profiles across studies can be achieved by using the same seed and sample size for each study in the “Model and Execution” window, without any filter.

runs used synchronous calling and the full persona representation. Table 1 reports input and output token use and the realized first-run cost per simulated respondent. Web appendix C describes the measurement procedure and provides detailed cost calculations.

TABLE 1: COST COMPARISON FOR FIVE DIGITAL-TWIN SURVEY REPLICATIONS

Study	Questions per twin	Answer units per twin	gpt-4o-mini		gpt-5.6 Luna (Medium)		gpt-5-mini (Medium)	
			Tokens per twin (input / output)	Cost per twin	Tokens per twin (input / output)	Cost per twin	Tokens per twin (input / output)	Cost per twin
Default Effects	2	2	37,442 / 132	\$0.0057	37,441 / 176	\$0.0077	37,441 / 679	\$0.0106
Idea Evaluation	11–12	11–12	39,076 / 460	\$0.0054	39,075 / 489	\$0.0084	39,075 / 1,304	\$0.0091
Promiscuous Donors	18	20	39,499 / 908	\$0.0050	39,498 / 646	\$0.0087	39,498 / 1,811	\$0.0086
Accuracy Nudges for Misinformation	30–31	30–31	40,514 / 1,254	\$0.0048	40,513 / 1,164	\$0.0095	40,513 / 3,316	\$0.0108
Fees Accuracy	63	63	46,604 / 2,553	\$0.0065	46,603 / 2,142	\$0.0119	46,603 / 4,560	\$0.0138

NOTE.—The ranges in question counts reflect differences in the number of questions displayed across between-subject conditions. All numbers come from an August 2026 replication. Output-token counts include reasoning tokens for the two reasoning models; `gpt-4o-mini` is not a reasoning model and therefore produces no reasoning tokens.

Evaluation Measures. We assess survey execution fidelity at the answer-unit level, using the first-run rate of structurally valid responses and the outcomes of the repair process. The first-run answered rate is the share of asked units that receive a nonblank response within the permitted range; we separately count forced-response blanks, out-of-range answers, and blanks on optional questions. For forced-response blanks and out-of-range answers, which we classify as fidelity failures, we also report how many are recovered through deterministic rematching or targeted reruns and how many remain unresolved.

Results

Fidelity Checks. Of 197,000 asked question–answer units, 99.60% received an in-range, nonblank response on the first run. Because this broad answered-rate measure treats optional blanks as unanswered, we distinguish among 207 forced-response omissions (0.11%), 67 out-of-range responses (0.03%), and 521 optional blanks (0.26%). Thus, 274 units (0.14%) were classified as survey execution failures. Of these failures, 95% occurred in two studies, Preferences for Redistribution and Heterogeneous Story Beliefs, and appeared to reflect long-context execution

errors in which the model lost track of the question sequence or returned an answer in the wrong format.

In the Preferences for Redistribution experiment, roughly 50 of 300 twins did not preserve a four-question sequence, placing a numeric response intended for a later question into an earlier binary-choice item and omitting the intervening questions. In the Heterogeneous Story Beliefs experiment, 24 twins left some forced belief questions unanswered, with one low-responding twin accounting for 33 skipped forced-response cells. The repair pipeline resolved 273 of the 274 survey execution failures: the rematching recovered 4 cells, and targeted reruns resolved another 268, and a real-time retry of a failed first-pass call recovered 1. After repair, only one forced-response item out of 197,000 remained unanswered. The 521 optional blanks were mainly end-of-survey comment boxes (“Do you have any comments about our survey?”). They are documented separately and are not classified as fidelity failures. Table A3 reports the complete study-level fidelity checks and repair outcomes.

GENERAL DISCUSSION

ExploraTwin reduces the cost and friction of running digital twin simulations by providing a standardized simulation pipeline, from survey construction or QSF upload through prompt generation, model execution, validation, repair, and clean data export. Our goal is to make it easier for the market research community to experiment with digital twin simulations, and help anyone develop their own empirical evidence related to their own use case. While ExploraTwin uses the Twin-2K-500 panel of digital twins by default, we also developed CroissantTwin, a standardized data format for adding samples of synthetic personas to the platform. The platform already provides access to 11 persona banks, and new banks can be easily added to the platform (or created for private use only).

We close by acknowledging limitations of the platform, which future research may address. First, a small share of twin responses can still violate the survey’s structure because the

LLM completes the instrument in context rather than through the rule-enforcing interface used by human respondents. Post-hoc repair retains the original responses, allowing researchers to inspect the errors. However, it may re-ask a question after the model has seen later parts of the survey, creating a different information state from the original survey flow. In our validation run, only 0.14% of answer units required repair, but moving validation into the response loop is a natural next step. Second, instrument support has boundaries. The pipeline currently accepts Qualtrics files, and the built-in builder targets straightforward linear surveys rather than complex logic. Features that depend on custom JavaScript, dynamic displays, games, or other interactive paradigms are detected and flagged but not executed. Studies using these features still require bespoke engineering. Extending the administration layer toward these interactive designs is an open direction. Third, the platform does not provide any estimate of the validity of the data it simulates. Rather, it is offered as a tool that makes it easier for anyone to test the validity of synthetic data themselves in their own context. But in light of known limitations of synthetic data and of some systematic distortions, future research could develop methods for quantifying the confidence researchers may place in the simulated results.

APPENDIX

POST-SIMULATION VALIDATION REPORT

Table A1 summarizes the six components of the validation report generated after each survey-mode run. Together, these components document the run configuration, survey fidelity, response validity, randomization, descriptive results, and coverage of the selected persona bank.

TABLE A1: POST-SIMULATION VALIDATION REPORT

Report section	What it documents
Experiment setting	The study name, run mode, sample size, persona bank, persona representation, model, token usage, API calls, panel filters, seed, question count, detected logic features, included and excluded blocks, and exported response files.
QSF fidelity	The parsed survey elements in the uploaded Qualtrics file, including question types, support levels, and warnings for features that are only partially supported or detected but not executed.
Response validity	Run-level checks for fully blank rows, fully skipped twins, forced-response omissions, duplicate response patterns, invalid or out-of-range answers, and the share of questions with no flags. Problems are flagged rather than silently corrected.
Randomization and balance	The survey's randomization rules and the realized number of twins assigned to each condition or randomizer arm, including whether assignment cells are balanced and which blanks are structural non-exposure.
Descriptive statistics	Question-level response summaries in survey order, with item text, response counts, missingness, validation flags, and visual distributions for each analyzable survey item.
Persona coverage	The demographic composition of the simulated panel, with thin cells flagged so the researcher can assess whether the realized panel is appropriate for the intended target population.

NOTE.—Overall status: A compact run-level status classifies the study as *ok*, *caution*, or *error* based on the share of problematic questions and failed calls. The status is a guide for review; the underlying flags remain visible.

PANEL MODE INTERFACES

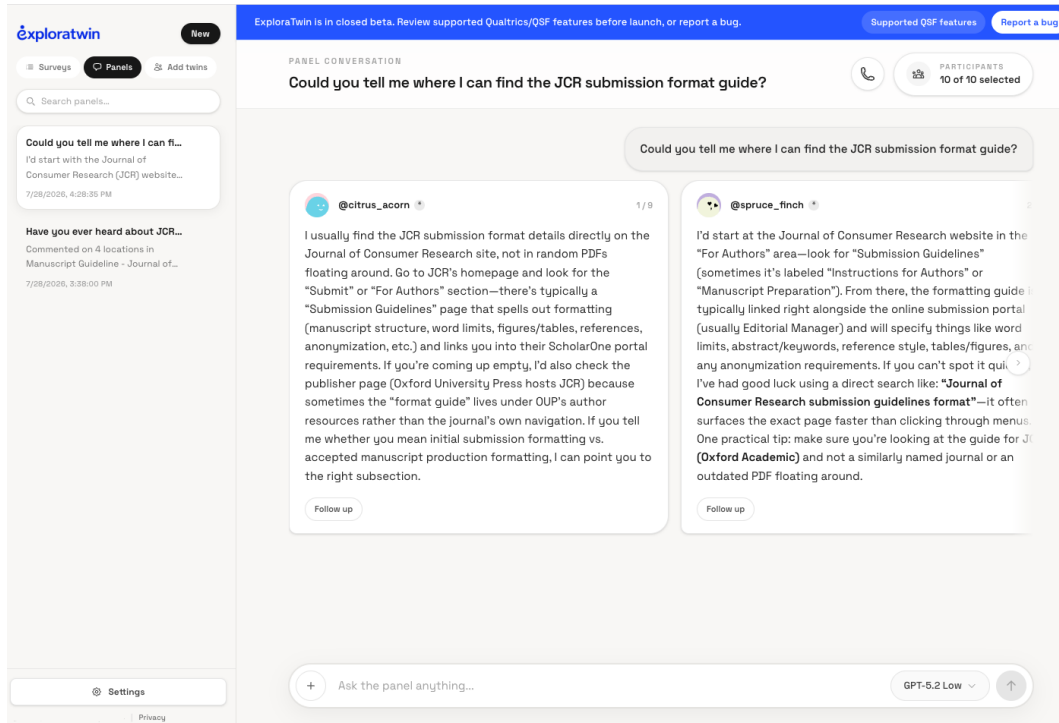


FIGURE A1: PANEL MODE: TEXT-BASED CONVERSATION

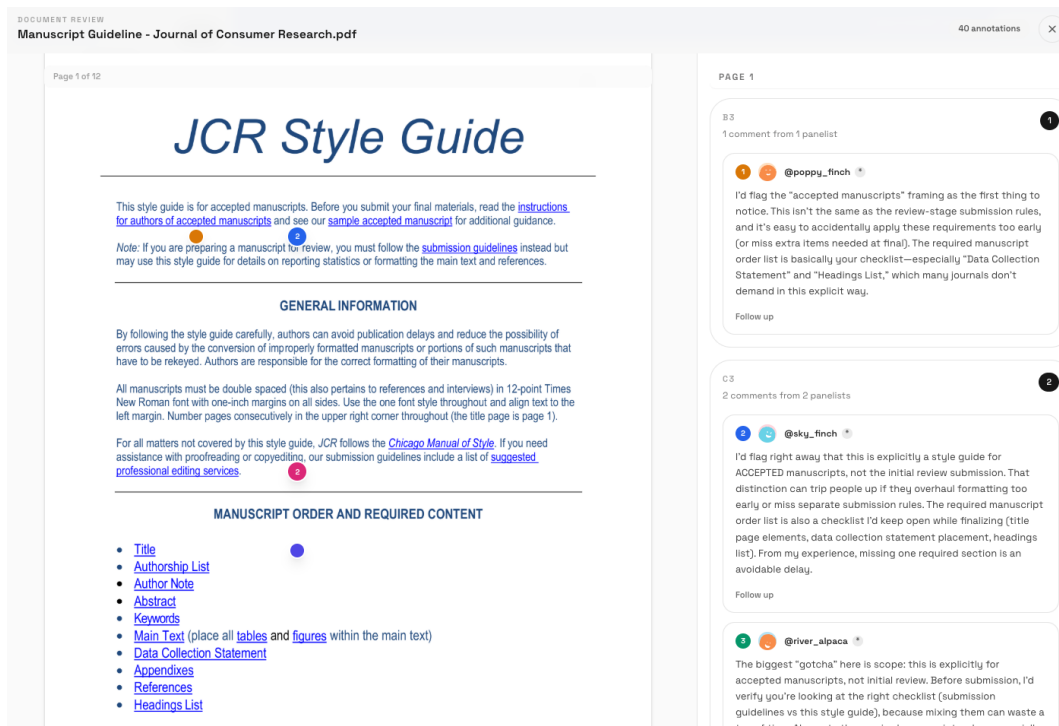


FIGURE A2: PANEL MODE: DOCUMENT REVIEW AND LOCATION-BASED ANNOTATION

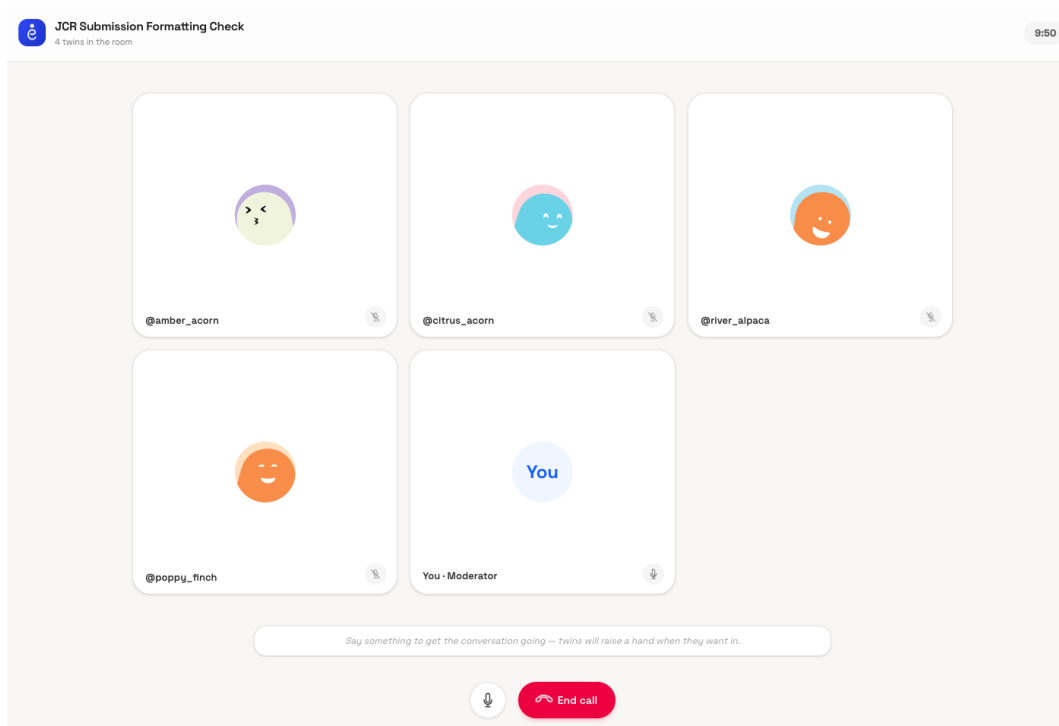


FIGURE A3: PANEL MODE: TWINMEET MODERATED VOICE INTERFACE

SURVEY SUPPORT AND SIMULATION FIDELITY

This appendix describes how ExploraTwin evaluates whether a simulation reproduces the study a researcher intended to field. The guiding principle is that the pipeline should preserve the survey’s intended exposure and response structure—or clearly flag features it cannot preserve. We organize this fidelity assessment into three layers of survey design and document the question types, flow features, and response structures that the platform currently supports.

Survey Layer

Most survey instruments can be understood as three connected layers. The first layer is the *question format*: the type of response task a participant faces, such as selecting one or multiple options, entering text, moving a slider, or ranking items. The second layer is the *question setting*: the local rules that shape how that question appears and how an answer is recorded, such as answer randomization, recoded values, default selections, response requirements and limits, and media attached to question text or answer choices. The third layer is *survey flow and logic*: the structure that determines which questions a participant sees and in what order, with examples such as blocks, randomizers, embedded data, branch logic, display logic, skip logic, loop and merge, and piped text. For supported question and logic features, ExploraTwin converts the survey design into an LLM-readable representation, presenting questions as structured text, translating logic rules into natural-language instructions, and including image inputs when visual stimuli are present. For example, a rule requiring respondents to choose two to three options becomes an explicit prompt instruction to choose two to three options.

Survey Design Support

ExploraTwin documents support at all three layers, using the categories and names from Qualtrics’ own survey-builder documentation so that a researcher can check their study feature

by feature. A feature is documented as *supported*, meaning it is parsed, administered to the twin, constrained, and exported; *partially supported*, meaning it runs with a documented simplification; or *not supported*, meaning the platform does not execute it. The design principle for the last category is that nothing is silently simulated: unsupported features are detected at upload, warned about before the run, skipped during simulation, and recorded in the validation report. The one exception is purely visual formatting (e.g., question layout, answer-choice styling, page breaks), which is listed as not reproduced but generates no warning, because it changes how a survey looks rather than what a respondent is asked. Table A2 summarizes the full support matrix.

Several boundary cases are especially important. Qualtrics allows researchers to attach custom JavaScript code that changes what a respondent sees or how a question behaves in the browser. ExploraTwin detects the presence of custom JavaScript and reports it, but does not execute it, because arbitrary client-side code can depend on timing, browser state, hidden page elements, external services, or side effects that are not recoverable from the QSF in a reliable way. The design principle is that unsupported or risky behavior should be made visible before a run. The same detection applies to the features in the “Not supported” column of table A2: inaccessible media, external web services, browser-only interactions, and unsupported specialty question formats. These features do not necessarily make a study unusable, but they change what can be claimed about the simulation, so the researcher should see them explicitly.

Table A2 presents the complete support matrix for Qualtrics survey features in ExploraTwin. The matrix organizes features by question format, question setting, and survey flow, and indicates whether each is fully supported, partially supported, or not supported. These classifications describe the platform’s current behavior and identify features that cannot yet be faithfully reproduced.

TABLE A2: SURVEY-FEATURE SUPPORT MATRIX

Survey design layer	Supported	Partially supported	Not supported (detected and warned)
Question format	• Multiple choice (single and multiple answer, drop-down, select box, write-in options) • Text entry (single line, multiline, essay, password) • Text / Graphic • Matrix table (Likert, Bipolar, Profile, text-entry rows, constant-sum rows) • Slider (sliders, bars, star ratings) • Number scale • Rank order • Side by side	• Form field • Constant sum • Matrix table (MaxDiff and rank-order variants) • Graphic slider (runs as a plain numeric slider, so the changing visual is lost) • Side-by-side columns with recoding or multiple answers	• Calendar • Net Promoter Score • Autocomplete • Pick, group, and rank • Hot spot • Heat map • Drill down • Highlight • Signature • File upload • Video response • Screen capture • Location selector • ArcGIS map • Tree testing • Unmoderated user testing • Interview selector • Qualtrics add-on project modules • Survey metadata (Timing, meta info, and captcha)
Question settings	• Response requirements (force and request response) • Response validation (choice counts, numeric ranges, character limits, rank uniqueness, constant-sum totals, numeric content types) • Recode values (multiple choice, matrix, rank order, constant sum) • Choice randomization, basic and advanced (fixed positions, random subsets, undisplayed choices) • Rich-content question text, with images passed to the twin as image input	• Display logic (embedded-data and common prior-answer conditions) • Skip logic (forward skips on selected-choice conditions) • Piped text (embedded-data fields fully, and prior answers and loop fields for common forms) • Default choices (shown to the twin as defaults, not forced) • Recode values on other question types (preserved with a warning) • Consistent-reversal randomization groups (preserved with a warning)	• Carry-forward choices and answers • Custom JavaScript • Custom validation beyond the standard rules • AI response-clarity validation • Math operations • Scoring • Translations (only the default language is used) • Piped values from date, GeoIP, location, scoring, quota, or contact-list sources • Visual formatting (question layout, answer-choice styling, page breaks, not warned)
Survey flow and block options	• Question blocks and block order • Groups • Embedded data defined in the survey • Randomizer (normal and even presentation, seeded per twin)	• Branch logic (embedded-data and common prior-answer conditions, executed in stages) • Question randomization within a block (basic shuffling, with advanced page and bucket modes warned) • Loop and merge (static loop tables, including random subsets, with dynamic sources not supported) • End of survey (early termination executed, with screen-out and redirect behavior not modeled)	• Authenticators • Quotas (tool and flow element) • Web service calls • Table of contents • Reference surveys • Supplemental data sources • Contact-list and panel state • Query-string parameters (values not contained in the survey file) • Branch conditions on quotas, scoring, device, or contact fields

Note. Question-type, setting, and flow labels follow Qualtrics’ official support documentation where possible. When ExploraTwin groups related variants, the table uses the closest Qualtrics term and describes the platform’s handling. Support categories reflect ExploraTwin’s parser and run-level validation report.

SIMULATION FIDELITY AND REPAIR

Table A3 reports first-run response validity and repair outcomes for the 19 replicated studies. Panel A distinguishes fidelity failures from blanks on optional questions, while panel B reports the flagged responses resolved through rematching and targeted reruns. Original responses are retained so that researchers can audit first-run performance.

TABLE A3: SIMULATION FIDELITY AND REPAIR RESULTS ACROSS 19 STUDIES

Panel A. First-run response validity

Study Name	Question Count	Answered Rate	F-SKIP	OPT	OOR
Accuracy Nudges for Misinformation	9,150	99.70%	0	27	0
Affective Primes	5,250	99.96%	0	0	2
Consumer Minimalism	4,800	99.98%	1	0	0
Context Effects	1,200	100.00%	0	0	0
Default Effects	600	100.00%	0	0	0
Digital Certificates for Luxury Consumption	3,000	100.00%	0	0	0
Hiring Algorithms	14,700	100.00%	0	0	0
Idea Evaluation	3,500	100.00%	0	0	0
Measures of Creativity	12,300	100.00%	0	0	0
Infotainment News Sharing	11,100	99.56%	0	49	0
Fees Accuracy	18,900	100.00%	0	0	0
Obedient Twins	3,600	100.00%	0	0	0
Preferences for Redistribution	8,400	97.46%	104	50	59
Privacy Preferences	1,800	100.00%	0	0	0
Promiscuous Donors	6,000	99.82%	0	7	4
Quantitative Intuition	18,900	99.96%	6	0	2
User Behavior with Recommendation Systems	41,100	100.00%	0	0	0
Heterogeneous Story Beliefs	31,800	98.48%	96	388	0
Targeting Fairness	900	100.00%	0	0	0
Total	197,000	99.60%	207	521	67

Panel B. Repair outcomes

Study Name	Answered Rate Change	Repaired Forced-Skips	Repaired Out-of-Range	Repaired
Accuracy Nudges for Misinformation	99.70% → 99.70%	0 → 0 (0)	0 → 0 (0)	0
Affective Primes	99.96% → 100.00%	0 → 0 (0)	2 → 0 (2)	2
Consumer Minimalism	99.98% → 100.00%	1 → 0 (1)	0 → 0 (0)	1
Context Effects	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Default Effects	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Digital Certificates for Luxury Consumption	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Hiring Algorithms	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Idea Evaluation	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Measures of Creativity	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Infotainment News Sharing	99.56% → 99.56%	0 → 0 (0)	0 → 0 (0)	0
Fees Accuracy	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Obedient Twins	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Preferences for Redistribution	97.46% → 99.41%	104 → 0 (104)	59 → 0 (59)	163
Privacy Preferences	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Promiscuous Donors	99.82% → 99.88%	0 → 0 (0)	4 → 0 (4)	4
Quantitative Intuition	99.96% → 100.00%	6 → 0 (6)	2 → 0 (2)	8
User Behavior with Recommendation Systems	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Heterogeneous Story Beliefs	98.48% → 98.78%	96 → 1 (95)	0 → 0 (0)	95
Targeting Fairness	100.00% → 100.00%	0 → 0 (0)	0 → 0 (0)	0
Total	99.60% → 99.74%	207 → 1 (206)	67 → 0 (67)	273

Note. Question count refers to asked question–answer units. Answered Rate is the share of units with an in-range, nonblank response; it therefore counts optional blanks as unanswered. F-SKIP indicates forced-question omissions, OPT indicates optional blanks, and OOR indicates out-of-range answers. Optional blanks are reported separately and are not classified as fidelity failures. Parentheses in the repaired columns report the number of cells resolved or recovered by the repair pass.

REFERENCES

- Aher, Gati V., Rosa I. Arriaga, and Adam Tauman Kalai (2023), “Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies,” in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, 337–71, <https://proceedings.mlr.press/v202/aher23a.html>.
- Akhtar, Mubashara, Omar Benjelloun, Costanza Conforti, et al. (2024), “Croissant: A Metadata Format for ML-Ready Datasets,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 82133–48, https://proceedings.neurips.cc/paper_files/paper/2024/file/9547b09b722f2948ff3ddb5d86002bc0-Paper-Datasets_and_Benchmarks_Track.pdf.
- American National Election Studies (2026), “ANES 2024 Time Series Study Full Release,” Last Accessed August 19, 2026. <https://electionstudies.org/data-center/2024-time-series-study/>.
- Argyle, Lisa P., Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate (2023), “Out of One, Many: Using Language Models to Simulate Human Samples,” *Political Analysis*, 31(3), 337–51, <https://doi.org/10.1017/pan.2023.2>.
- Ashokkumar, Ashwini, Luke Hewitt, Isaias Ghezze, and Robb Willer (2026), “Large Language Models Can Predict the Results of Social Science Experiments,” *Nature*, 656, 115–22, <https://doi.org/10.1038/s41586-026-10742-x>.
- Ge, Tao, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu (2024), “Scaling Synthetic Data Creation with 1,000,000,000 Personas,” <https://arxiv.org/abs/2406.20094>.
- Manning, Benjamin S. and John J. Horton (2026), *General Social Agents*, NBER Working Paper 34937, National Bureau of Economic Research, <https://doi.org/10.3386/w34937>.
- National Center for Health Statistics (2024), “National Health and Nutrition Examination Survey, August 2021–August 2023,” Last Accessed August 19, 2026. <https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?Cycle=2021-2023>.

NVIDIA Corporation (2026), “Nemotron-Personas,” Last Accessed August 19, 2026. <https://huggingface.co/collections/nvidia/nemotron-personas>.

Park, Joon Sung, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein (2023), “Generative Agents: Interactive Simulacra of Human Behavior,” in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, ed. Sean Follmer, New York, NY: Association for Computing Machinery, 1–22, <https://doi.org/10.1145/3586183.3606763>.

Park, Joon Sung, Carolyn Q. Zou, Jonne Kamphorst, et al. (2024), “LLM Agents Grounded in Self-Reports Enable General-Purpose Simulation of Individuals,” <https://arxiv.org/abs/2411.10109>.

Peng, Tianyi, George Gui, Melanie Brucks, et al. (2025), “Digital Twins as Funhouse Mirrors: Five Key Distortions,” <https://arxiv.org/abs/2509.19088>.

Thongpramoon, Pongsasit, Verdi March, Christopher Low, Shyamala Prayaga, Dane Corneil, and Yev Meyer (2026), “Nemotron-Personas-Singapore: Synthetic Personas Aligned to Real-World Distributions for Singapore,” Last Accessed August 19, 2026. <https://huggingface.co/datasets/nvidia/Nemotron-Personas-Singapore>.

Toubia, Olivier, George Z. Gui, Tianyi Peng, Daniel J. Merlau, Ang Li, and Haozhe Chen (2025), “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*, 44(6), 1446–55, <https://doi.org/10.1287/mksc.2025.0262>.

Wang, Ziyi, Yuxuan Lu, Wenbo Li, et al. (2026), “OPERA: A Dataset of Observation, Persona, Rationale, and Action for Evaluating LLMS on Human Online Shopping Behavior Simulation,” in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, San Diego, CA: Association for Computational Linguistics, 43942–60, <https://aclanthology.org/2026.acl-long.2033/>.

WEB APPENDIX

Brief Commentary: ExploraTwin, a Non-Profit Research Platform for Digital Twin Simulations

This document contains three web appendixes supporting the ExploraTwin platform and its validation. Web appendix A describes response validation, rematching, and targeted reruns; web appendix B presents the CroissantTwin specification and Build Skill; and web appendix C documents cost accounting and estimation.

WEB APPENDIX A

RESPONSE VALIDATION, REMATCHING, AND TARGETED RERUNS

This web appendix describes how ExploraTwin identifies structurally problematic responses and how researchers can repair them after a simulation. These procedures address execution and formatting errors. For format errors, ExploraTwin provides a rematch mechanism that lets the researcher map the twin’s existing answer back to a valid survey option when the mapping is defensible. Rematching is rule-based: the pipeline first applies deterministic normalizations, such as case, spacing, punctuation, and label–value equivalences, and then uses fuzzy string matching to map the returned text to the closest valid option. Answers without a close match remain flagged. For missing or out-of-range answers, the researcher can optionally rerun the problematic twin. The rerun pipeline preserves the twin’s valid answers from the original run and explicitly re-asks the problematic questions in context.

Initial Response Validation

ExploraTwin distinguishes two types of responses that may require repair. An *out-of-range response* does not correspond to an offered option or falls outside the permitted numeric range. This can occur when a twin paraphrases an option, includes additional text, or returns an answer in the wrong format. A *forced skip* occurs when a forced-response question is left blank. Blanks on optional questions are documented separately and do not enter the repair process. Noticeably, ExploraTwin distinguishes call-level failures from response-level errors. If an API call fails during execution because of a provider or network error, the platform automatically retries the call as part of the initial run; post-hoc repair is used only after a response has been returned but is missing, out of range, or structurally invalid.

During the initial simulation, every closed-ended response is checked against the survey

template by a deterministic option matcher. The matcher accommodates format-level variation by normalizing capitalization, spacing, punctuation, and Unicode characters; removing leading answer codes; recognizing reordered labels such as “Agree strongly” and “Strongly agree”; and mapping embedded numbers to numeric options or labeled ranges. A rule is applied only when it identifies a unique valid option. Fuzzy matching is not used during the initial run.

When the matcher cannot identify a valid option, ExploraTwin preserves the model’s verbatim response, exports the corresponding cell as blank, and flags it as out of range. A missing forced response is similarly flagged as a forced skip. The Participant Results view allows researchers to filter for these cells, compare the original response with the expected options, and select one of two repair procedures. Rematching attempts to recover an existing answer without another model call. Targeted rerunning re-asks unresolved questions. Both procedures write to a separate repaired dataset and leave the first-run responses unchanged.

Rematching

Rematching is a rule-based procedure for flagged cells in which the twin returned an answer that may correspond to a valid option. The platform first compares the preserved response with the option labels while ignoring capitalization. If no match is found, it reapplies the deterministic normalizations used during the initial simulation. As a final step, it calculates the character-level similarity between the response and each valid option and proposes the closest option only when the similarity score is at least 0.60.

Character similarity can be misleading when labels are textually similar but semantically opposed. The platform therefore blocks a fuzzy proposal when one label contains a negation that the other lacks, when similar word stems carry opposing prefixes, or when the labels belong to a known conflicting pair. These checks distinguish, for example, “satisfied” from “dissatisfied,” “increased” from “decreased,” and “Male” from “Female.”

The proposed mapping and the rule that produced it are displayed in the interface. Re-

searchers may accept or reject proposals individually or review proposed mappings in bulk. Accepted mappings are written to the repaired dataset together with the matching rule. Answers without a defensible match remain flagged and can be included in a targeted rerun. Fuzzy matching is therefore confined to a researcher-controlled repair step and is never applied silently during the original simulation.

Targeted Reruns

Targeted rerunning is available for forced skips and out-of-range responses that rematching cannot resolve. Researchers may select one twin or all twins with unresolved flags; the platform does not rerun unflagged panel members. Before making any model calls, the website displays the estimated repair cost for confirmation. It then makes one new call for each selected twin.

The rerun uses the twin's original persona representation and system instruction. It presents the survey in its original order with the twin's prior valid answers retained, while marking only the unresolved cells as requiring an answer. The model is instructed to answer those cells and leave all previously valid responses unchanged. This design prevents the repair process from resampling valid responses after the researcher has observed them.

Each new answer is evaluated by the same deterministic matcher used during the initial simulation. If the response still cannot be mapped to a valid option, it remains flagged rather than being forced into the dataset. The export records the repair method for each repaired cell and retains the raw and repaired response files side by side.

Because targeted rerunning occurs after the original simulation, a repaired response is generated in a different information state from an answer produced at its original position in the survey flow. Preserving the first-run data allows researchers to inspect this distinction and conduct sensitivity checks when sequential exposure is substantively important. The manuscript appendix reports the study-level flags and repair outcomes from the 19-study assessment.

WEB APPENDIX B

THE CROISSANTTWIN SPECIFICATION AND BUILD SKILL

CroissantTwin (CT) is an application profile for packaging persona banks as portable, inspectable, and provenance-aware datasets. It extends the MLCommons Croissant 1.1 standard rather than defining a new file format. Croissant supplies the general mechanisms for describing datasets, files, schemas, typed fields, joins, checksums, and provenance. CT adds the concepts required specifically for persona banks, including personas, alternative representations of each persona, persona basis, approved filter fields, and disclosures concerning access and human-derived data. The current implementation targets the versioned CT 0.1 working specification.

The CroissantTwin Specification

The CroissantTwin specification is a public rulebook defining what a portable persona bank must contain and how its components are described. It replaces dataset-specific structures with a common, machine-readable structure, allowing compatible tools to use a bank without custom code. Because the rules are public and platform-independent, researchers can create conformant banks and developers can build tools that use them without relying on ExploraTwin.

The specification distinguishes among three components. The *bank* is the citable and versioned dataset as a whole. A *Persona* record identifies the stable unit being simulated and stores the authoritative attributes used for filtering. A *Representation* is a particular version of that persona's information supplied to the model, such as a full history, summary, or demographics-only profile. This separation allows the same persona to have multiple representations without duplicating its identity or filter information.

Each conformant bank includes a machine-readable manifest, conventionally named `croissant.json`. The manifest identifies the Persona and Representation records, explains how they are linked, de-

clares the available filters, and locates the representation content. The underlying data can remain in common formats such as CSV, JSON, JSON Lines, Parquet, plain text, or Markdown, and existing files can be referenced directly when they already satisfy the required structure.

The manifest also documents whether the personas represent observed individuals, human composites, synthetic personas, or a mixture; as well as the bank's provenance, license, access conditions, intended uses, and limitations. Filters must be explicitly approved, and identifiers, unstructured text, and sensitive attributes are not exposed by default. Conformance describes a bank's structure and disclosures; it does not certify consent, anonymity, representativeness, legal compliance, or similarity to human respondents.

The Build Skill

The Build Skill is a self-contained workflow for coding agents that converts an unfamiliar dataset and its associated paper or documentation into a CT bank. It does not depend on a pre-installed CT command-line tool, software library, service, or registry. Instead, the agent inspects the source, proposes a documented conversion plan, and writes source-specific transformation and validation code after the researcher confirms the plan.

Source Inspection and Evidence. The workflow begins by treating the source directory as immutable. The agent inventories its files, formats, sizes, checksums, schemas, and record counts; identifies candidate participant, study, experiment, session, and wave identifiers; and examines repeated identifiers, missing-value codes, candidate filters, sensitive fields, and possible representation content. The inspection supports CSV, JSON, JSON Lines, Parquet, text, and Markdown sources.

The associated paper or documentation is then used to interpret the files. The agent records evidence concerning the unit of observation, sample, recruitment, experiment structure, repeated measurements, identifier scope, variable meanings, data collection, persona origin, and stated

limitations. The workflow also does not treat an inference as a documented fact and does not infer a dataset license from the publication license of its paper.

Persona and Representation Decisions. The agent next reconciles the paper with the observed files and proposes the smallest persona model that preserves the dataset’s meaning. If one confirmed identifier follows the same participant across experiments, the default proposal is one Persona whose ordered experimental records form a behavioral-history representation. If identifiers are only unique within a study or experiment, the agent retains scoped personas rather than inventing links across files.

The conversion plan identifies the persona identifier, grouping and ordering rules, canonical Persona fields, approved filters, representation kinds, payload format, missing-value treatment, persona basis, license, and access conditions. Before transformation, the agent presents one example of the representation text that would be supplied to the language model. It also reports the anticipated distribution of representation lengths and compares the largest payload with the declared context budget (100,000 tokens per payload). Data cannot be silently discarded to satisfy the budget.

Persona identity, cross-study linkage, representation design, filter exposure, persona basis, provenance, licensing, and access are treated as blocking decisions. The agent proceeds only after the researcher confirms the complete conversion plan and rendered prompt example.

Building the Persona Bank. After confirmation, the agent records the approved decisions in a machine- readable import mapping and writes a source-specific transformation program. The program applies the confirmed identifier, grouping, ordering, missing-data, filtering, and representation rules while leaving the original files unchanged. It produces a publishable bank containing the manifest, bank card, Persona records, and Representation records. Separately, it retains the confirmed mapping, transformation code, and validation results needed to audit or reproduce the conversion.

Validation and Upload Readiness. Finally, the agent validates both the manifest and the materialized records. It checks identifier uniqueness, persona–representation links, representation coverage and payload modes, declared fields and codebooks, approved filters, access and license documentation, file paths and checksums, and the size of prompt-facing representations. When practical, it repeats the transformation and compares the outputs to assess reproducibility. The skill reports whether the resulting package is locally ready for upload, but it does not publish or send the bank to ExploraTwin without explicit authorization. The validated bank is then packaged as a single .zip archive containing only the bank directory’s contents, excluding the source data and build artifacts; this archive is the workflow’s deliverable and the file a researcher uploads to ExploraTwin.

WEB APPENDIX C

COST ACCOUNTING AND ESTIMATION

This web appendix documents how the costs reported in the main text were measured and how the estimated columns of the cost-comparison table were computed.

API Execution. Execution mode depended on the survey’s logic and flow. Twelve studies whose survey flow could be determined in advance were submitted through the OpenAI Batch API. Six studies containing answer-dependent logic were executed synchronously because the platform had to observe each twin’s responses before determining what to present next. One additional static study, Infotainment News Sharing, was ultimately executed in real time because its submitted batch did not complete within the pipeline’s two-hour waiting limit. Thus, seven studies were executed synchronously. Among these seven studies, three required two sequential calls per twin to administer multiple stages; the other four were completed in one call per twin. Across execution modes, the model, persona representation, prompts, and survey-administration procedures remained the same. Successful runtime retries are included in the first-run cost, whereas post-hoc reruns conducted to repair returned responses are reported separately as repair costs.

Cost Computation. We calculated the cost of each call by multiplying its provider-reported input and output token consumption by the applicable API prices. Reasoning tokens—internal tokens used by the model before producing its visible response—are billed as output tokens and are included in the reported output-token counts. They accounted for 62% of all output tokens in the 19-study replication. Because the persona representation constitutes most of the input tokens in each call, adding questions within the same call generally adds relatively few tokens and therefore has a small marginal cost.

TABLE C1: TRACKED COSTS FOR THE 19 DIGITAL-TWIN REPLICATIONS

Study	Questions per twin	Answer units per twin	Avg. tokens per twin (input / output)	Cost per twin: gpt-5-mini	Primary calling mode	Repair calls	Repair cost per repaired twin
Accuracy Nudges for Misinformation	30–31	30–31	40,513 / 3,358	\$0.0084	Batch	0	—
Affective Primes [†]	5	13–22	74,686 / 2,670	\$0.0187	Real time	1	\$0.0142
Consumer Minimalism	5	16	39,575 / 3,035	\$0.0080	Batch	0	—
Context Effects	4	4	38,592 / 847	\$0.0048	Batch	0	—
Default Effects	2	2	37,441 / 643	\$0.0044	Batch	0	—
Digital Certificates for Luxury Consumption [†]	8	10	72,366 / 2,677	\$0.0105	Real time	0	—
Fees Accuracy	63	63	46,603 / 4,683	\$0.0150	Real time	0	—
Heterogeneous Story Beliefs	43	106	54,528 / 8,960	\$0.0266	Real time	24	\$0.0502
Hiring Algorithms	21	49	47,970 / 4,547	\$0.0090	Batch	0	—
Idea Evaluation	11–12	11–12	39,075 / 1,249	\$0.0047	Batch	0	—
Infotainment News Sharing	29	37	42,203 / 4,137	\$0.0188	Real time	0	—
Measures of Creativity	8	41	41,125 / 2,896	\$0.0067	Batch	0	—
Obedient Twins [†]	12	12	73,052 / 2,160	\$0.0106	Real time	0	—
Preferences for Redistribution	28	28	40,924 / 3,043	\$0.0082	Batch	57	\$0.0108
Privacy Preferences	6	6	36,611 / 1,180	\$0.0052	Batch	0	—
Promiscuous Donors	18	20	39,498 / 1,848	\$0.0066	Real time	0	—
Quantitative Intuition	10	63	42,529 / 5,793	\$0.0111	Batch	7	\$0.0142
Targeting Fairness	3	3	36,230 / 908	\$0.0046	Batch	0	—
User Behavior with Recommendation Systems	4	137	54,005 / 7,618	\$0.0134	Batch	0	—
Total (19 studies)						89	\$1.9355 total

NOTE.—Ranges reflect between-condition differences in displayed questions. The primary calling mode indicates whether a study ran primarily in batch or real time. [†]These studies are administered in two passes because later questions depend on earlier answers; token counts and per-twin costs combine both passes. Repair calls report the number of twins rerun after the first-pass simulation.

Observed Costs and Results. Each model request generated a billing record containing the provider-reported input, output, and reasoning token counts and the corresponding dollar cost. The reported costs are therefore observed rather than estimated. The first-run simulations consumed 287.9 million tokens and cost \$58.55. Subsequent 89 repair calls cost an additional \$1.94, producing a total cost of \$60.49.

Table C1 reports the complete cost accounting for all 19 studies. Questions per twin count distinct base questions, whereas answer units count the resulting question–answer cells; a matrix or multi-select question can therefore contribute multiple answer units. Token counts and costs are reported per simulated respondent. Fifteen studies issued exactly one model call per twin, so cost per twin equals cost per call; the three studies administered in two passes issue two calls per twin, and their token counts and per-twin costs combine both passes. Heterogeneous Story Beliefs recorded 301 calls for 300 twins, so its two costs differ slightly.

Tracked per-twin costs range from \$0.0044 for Default Effects, with two answer units, to \$0.0266 for Heterogeneous Story Beliefs, with 106 answer units. No study exceeded \$8 in first-run cost for 300 twins.

Five-Study Cross-Model Cost Comparison

The cross-model comparison was conducted separately from the 19-study fidelity replication. We selected five studies spanning 2 to 63 displayed questions and reran each study with 300 twins using gpt-4o-mini, gpt-5.6 Luna with medium reasoning effort, and gpt-5-mini with medium reasoning effort. All runs used the full persona representation and synchronous API calls only (more expensive than batch API calling).

For each of the 15 model–study combinations, we recorded the provider-reported input and output tokens and realized dollar cost. Each model completed 1,500 simulated respondents across the five studies. The total first-run costs were \$8.22 for gpt-4o-mini, \$13.84 for gpt-5.6 Luna, and \$15.86 for gpt-5-mini. The average first-run cost for gpt-5-mini was \$0.0106 per

respondent in this comparison, compared with \$0.0078 across the same five studies in the 19-study replication. This difference partly reflects execution mode: all cross-model runs used synchronous calls, whereas three of the five studies used the lower-priced Batch API in the 19-study replication; differences in realized token consumption also contributed.

For the two reasoning models, output-token counts include reasoning tokens. Because `gpt-4o-mini` is not a reasoning model, it reports no reasoning tokens. The input and output token counts and realized first-run cost per simulated respondent are reported in Table 1 in the main text.