Technical Methods Documentation

Behavioral Experiment Simulation Tool v1.0.1.4

Proprietary Software | Dr. Eugen Dimant

(c) 2026 Dr. Eugen Dimant. All rights reserved.

1. Response Generation Framework

1.1 Persona-Based Modeling

The simulation employs a persona-based approach to model individual differences in survey responding. Each simulated participant is assigned to a response persona that determines their behavioral patterns throughout the survey.

Persona Distribution Parameters

Persona	Base Weight	Empirical Source
Engaged Responder	0.30-0.35	Krosnick (1991)
Satisficer	0.20-0.25	Krosnick (1991)
Extreme Responder	0.08-0.12	Greenleaf (1992)
Acquiescent Responder	0.06-0.08	Billiet & McClendon (2000)
Careless Responder	0.03-0.08	Meade & Craig (2012)
Socially Desirable Responder	0.10-0.15	Paulhus (1991)

Persona weights are adjusted based on study context. For instance, technology-focused studies increase the weight of tech-related personas; consumer studies adjust for relevant demographic patterns.

1.2 Trait Sampling

Each persona specifies probability distributions for response traits:

* Response tendency (0-1): Base position on response scales

* Ex

Individual participants sample trait values from persona-specific distributions, producing realistic within-persona variation.

2. Likert Scale Response Algorithm

Scale responses are generated through a multi-stage process:

Stage 1: Base Response

```
base = response_tendency x (scale_max - scale_min) + scale_min
```

Stage 2: Domain Calibration

Adjustments based on construct-specific norms:

*

Stage 3: Experimental Effect Application

Condition effects are applied based on semantic parsing of condition names:

```
effect = semantic_effect x configured_d x SCALE_FACTOR
```

The semantic parser identifies valence from condition labels (e.g., "high" vs. "low", "positive" vs. "negative") and applies effects accordingly.

Stage 4: Reverse Coding Handling

For reverse-coded items:

```
response = scale_max - (response - scale_min)
acquiescence_artifact = acquiescence x 0.25 x scale_range
```

Stage 5: Within-Person Variance

```
SD = (scale_range / 4) x consistency_trait
response += N(0, SD)
```

Stage 6: Extreme Response Style

```
if U(0,1) < extremity x 0.45:
response = endpoint (min or max based on current value)
```

Stage 7: Acquiescence Bias

```
response += (acquiescence - 0.5) x scale_range x 0.20
```

Stage 8: Social Desirability

```
response += (SD_trait - 0.5) x scale_range x 0.12  [for positive items]
```

Stage 9: Boundary Enforcement

```
response = round(clamp(response, scale_min, scale_max))
```

Target Distributional Properties:

*

3. Effect Size Implementation

3.1 Semantic Condition Parsing

Effect direction is determined by parsing condition names for semantic content:

Positive valence indicators: high, good, positive, friend, reward, benefit, gain, success, win, advantage

Negative valence indicators: low, bad, negative, enemy, punishment, cost, loss, failure, disadvantage

Domain-specific adjustments:

*

3.2 Effect Magnitude Calibration

Effect sizes follow Cohen's (1988) conventions with empirical calibration:

Category	Cohen's d	Scale Points (7-pt)
Small	0.20	~0.3 points
Medium	0.50	~0.7 points
Large	0.80	~1.1 points

The meta-analytic average for social psychology experiments is d = 0.43 (Richard et al., 2003), suggesting that "medium" effects represent typical experimental findings.

4. Survey Flow Logic

4.1 Question Visibility Determination

The simulation respects the survey's programmed logic. Questions are only presented to participants whose condition assignment would make those questions visible.

Detection Methods:

1. Explicit condition restrictions: Question metadata specifies visible conditions

2. D

4.2 Factorial Design Support

For crossed factorial designs (e.g., "AI x Hedonic"), the system:

*

5. Open-Ended Response Generation

5.1 Two-Tier Architecture

The system uses a primary AI-powered generator with a template-based fallback.

Primary (LLM-Powered):

1. Batch prompt construction: For each question x condition x sentiment bucket, a prompt is built that includes study context, experimental condition, and N participant profiles (each specifying verbosity, formality, engagement, and sentiment)

2. L
C

Fallback (Template-Based):

1. Question type classification: 40+ types identified via regex patterns

2. D

5.2 Deep Variation Pipeline (7 Layers)

Each base response passes through independent transformation layers:

Layer	Transformation	Probability
0	Word micro-variation (drop, insert, s	30-50% per axis
1	Sentence restructuring (shuffle, relo	55% combined
2	Verbosity control (truncate or elabo	Persona-driven
3	Formality adjustment (casual starte	Persona-driven
4	Engagement modulation (truncate -	Persona-driven
5	Typo injection (realistic misspellings	25% for casual
6	Synonym swaps (1-3 per response)	Always
7	Punctuation variation	30%

Target uniqueness: >=84% unique text from a single base response at n=500; >=90% from a pool of 30 at n=2,000.

5.3 Smart Pool Scaling

Pool size per sentiment bucket: max(30, min(80, floor(sqrt(participants_per_bucket) x 3) + 10))

Where participants_per_bucket = sample_size / (n_conditions x 5 sentiments).

5.4 Multi-Provider Failover

Priority	Provider	Model	Rate Limit (Free)

1	Google AI Studio	Gemini 2.5 Flash Lite	Built-in key
2	Google AI Studio	Gemma 3 27B	Built-in key
3	Groq	Llama 3.3 70B	14,400 req/day
4	Cerebras	Llama 3.1 8B	1M tokens/day
5	OpenRouter	Llama 3.3 70B	Free models

Providers are tried in priority order; if one fails or rate-limits, the next is attempted automatically. Users can optionally provide their own API keys for higher rate limits.

5.5 Response Characteristics by Persona

Persona	Typical Length	Style	Quality
Engaged	2-4 sentences	Detailed, thoughtful	High
Satisficer	1 sentence	Brief, generic	Adequate
Careless	1-3 words	Minimal, off-topic	Poor
Extreme	1-2 sentences	Emphatic	Moderate

6. Exclusion Criteria Simulation

The simulation generates realistic exclusion flags:

Criterion	Implementation
Completion time	Based on attention level; careless = fas
Attention check failures	Probability = 1 - attention_level
Straight-lining	Consecutive identical responses based on

Flags combine into an overall exclusion recommendation based on configurable thresholds.

7. Reproducibility

7.1 Seeding Strategy

All random elements use seeded generators:

*

7.2 Cross-Platform Consistency

MD5 hashing (rather than Python's native hash()) ensures identical results across platforms and sessions.

8. Natural Language Design Parser (v1.3)

8.1 Condition Detection Pipeline

The conversational builder parses experiment descriptions using a multi-pattern matching approach:

1. Labeled parenthetical factorial: `N (Factor: level vs level) x N (Factor: level vs level)` -- Matches academic-style condition descriptions. Factors are crossed to produce all NxM conditions.
2. Explicit NxM factorial: `2x2`, `3x2` -- Detects dimensions from multiplication notation, then extracts factor names and levels from surrounding context.
3. Simple enumeration: `"Condition 1 vs Condition 2 vs Condition 3"` -- Splits on "vs", commas, numbered lists, or semicolons.
4. Trailing noise stripping: Removes non-condition text like "between-subjects, random assignment" before parsing.

8.2 Scale Parsing Pipeline

Scale descriptions are split into segments using prioritized delimiters:

1. Paragraph breaks (double newlines) -- most reliable for multi-paragraph input

2. N

Each segment is parsed for: scale name (from Name (Abbrev): ... pattern), number of items, scale range (N-point, min-max), type (likert/slider/numeric/binary), anchors, and reverse-coded items. Known validated instruments (BFI-10, PANAS, GAD-7, PHQ-9, etc.) are matched by abbreviation and auto-populated with canonical parameters.

8.3 Design Validation

The parser validates the complete design against:

*

9. References

Billiet, J. B., & McClendon, M. J. (2000). Modeling acquiescence in measurement models. Structural Equation Modeling, 7(4), 608-628.

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences (2nd ed.). Lawrence Erlbaum.

Dietvorst, B. J., Simmons, J. P., & Massey, C. (2015). Algorithm aversion. Journal of Experimental Psychology: General, 144(1), 114-126.

Greenleaf, E. A. (1992). Measuring extreme response style. Public Opinion Quarterly, 56(3), 328-351.

Krosnick, J. A. (1991). Response strategies for coping with the cognitive demands of attitude measures. Applied Cognitive Psychology, 5(3), 213-236.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. Academy of Management Review, 20(3), 709-734.

Meade, A. W., & Craig, S. B. (2012). Identifying careless responses in survey data. Psychological Methods, 17(3), 437-455.

Oliver, R. L. (1980). A cognitive model of the antecedents and consequences of satisfaction decisions. Journal of Marketing Research, 17(4), 460-469.

Paulhus, D. L. (1991). Measurement and control of response bias. In J. P. Robinson et al. (Eds.), Measures of Personality and Social Psychological Attitudes (pp. 17-59). Academic Press.

Richard, F. D., Bond, C. F., & Stokes-Zoota, J. J. (2003). One hundred years of social psychology quantitatively described. Review of General Psychology, 7(4), 331-363.

© 2026 Dr. Eugen Dimant. All rights reserved. Proprietary and confidential.