Supplemental Materials to Disentangling the Multifaceted Nature of Certainty in Evaluations.

Julian Quandt[1, 2], Bernd Figner[2], Rob W. Holland[2], Maria Teresa Carere[2], Marijn Eversdijk[2,3], and & Harm Veling[2, 4]

[1] WU Vienna University of Economics and Business

[2] Behavioural Science Institute

Radboud University

[3] Tilburg University

[4] Consumption and Healthy Lifestyles

Wageningen University and Research

Author Note

All preregistrations, materials and data are openly available on the Open Science Framework (OSF) under https://osf.io/9y38n/.

Correspondence concerning this article should be addressed to Julian Quandt, Welthandesplatz 1, D5, 1020 Vienna, Austria. E-mail: julian_quandt@live.de

Supplemental Materials to Disentangling the Multifaceted Nature of Certainty in Evaluations.

Contents

Details on Simulation in Figure 1

Left Panel: Individual Object Evaluations and Certainty

The left panel of Figure 1 illustrates how evidence sample consistency could underlie the apparent relation between evaluation certainty and evaluation extremity.

We simulated evaluations for 1000 objects i provided by a hypothetical participant. First, we simulate $n_{obs} = 100$ value-relevant experiences sampled from memory that are restricted to a [-1, 1] scale, resulting in an object evaluation μ_i. The samples are drawn from a transformed Beta distribution, as it provides a convenient but unrestrictive distribution to model bounded data. Given the constraint of the scale, we derive the maximum permissible variance $\sigma^2_{max_i}$ of the samples based on their mean μ_i, assuming that μ_i represents the observed evaluation. In other words, assuming that the evidence samples themselves entirely fall inside the same scale as the evaluation, the maximal variance of the sample $\sigma^2_{max_i}$ that can result in a given evaluation μ_i is restricted by:

$$\sigma^2_{max_i} = 1 - \mu_i^2. \tag{1}$$

At the same time, we restrict the minimum permissible variance of any sample mean μ across the entire scale to

$$\sigma^2_{min} = 1 - 0.99^2 \tag{2}$$

reflecting the maximum variance that a sample of $\mu_i = 0.99$ or $\mu_i = -0.99$ could have. Them we simulate a the samples from a Beta distribution:

$$X_i \sim Beta(a, b), X_i \in [0, 1] \tag{3}$$

and transform them to the [-1, 1] scale as

$$\hat{X}_i = 2X_i - 1, \hat{X}_i \in [-1, 1] \tag{4}$$

reflecting both negative- and positive-value evidence samples.

Specifically, for each evaluation μ_i we select a target variance $\sigma^2_{target_i}$ from a uniform distribution between σ^2_{min} and $\sigma^2_{max_i}$, reflecting that the variance of an evidence sample can fall anywhere in the permissible range. We then solve for the Beta distribution parameters α and β that result in the target variance $\sigma^2_{target_i}$ for the transformed samples $\hat{X}_i$, satisfying the requirements of

$$\mathbb{E}[X_i] = \frac{\mu_i + 1}{2}; Var(2X_i - 1) = 4Var(X_i) = \sigma^2_{target_i}. \tag{5}$$

Hence, we want $Var(X_i) = \frac{\sigma^2_{target_i}}{4}$. Reparametrizing the Beta distribution as

$$X \sim Beta(mS, (1-m)S), S = \alpha + \beta, m = \frac{\mu_i + 1}{2} \tag{6}$$

and hence defining

$$Var(X_i) = \frac{m(1-m)}{S+1} \tag{7}$$

and solving for S

$$S = \frac{4m(1-m)}{\sigma^2_{target_i}} - 1 \tag{8}$$

and finally transforming $\hat{X}_i = 2X_i - 1$ we create an experience sample $\hat{X}_i$ containing 100 experiences for $i = 1000$ equally spaced evaluations of $\mu_i \in [-.95, .95]$ and target variance $\sigma^2_{target_i}$. The choice of 100 experiences is arbitrary and only serves to illustrate the concept.

We then calculate the observed sample means μ_{obs_i} and posterior variance of the sample mean $\frac{\sigma^2_{obs_i}}{n_{obs}}$, where $n_{obs} = 100$ (i.e., we assume a flat prior and normally distributed μ_{obs_i}) for each $\hat{X}_i$. From this we calculate the sample consistency as the negative logarithm of the posterior variance

$$c_{1_i} = -\log(\max(\frac{\sigma^2_{obs_i}}{n_{obs}}, \epsilon_1)) \tag{9}$$

where ϵ is a small constant to avoid taking the log of zero. Mathematically, this is equivalent to the inverse of the posterior variance of the sample mean, but we use the logarithm for numerical stability with small values and for more convenient plotting.

Lastly, assuming that consistency c_{1_i} determines certainty c_{2_i}, but might additionally be influenced by other, unobserved factors and measurement error, we add a Gaussian noise term ϵ_2 to the consistency c_{1_i}, where the noise term is

$$\epsilon_2 \sim N(0, \sigma_{C_1} * \sqrt{\frac{1}{3}}), \tag{10}$$

such that explained variance R^2 of certainty by consistency is $\approx .75$. Note that the exact size of ϵ_2 is not theoretically motivated here but the argument would hold also for larger values of ϵ_2.

Right panel: Resulting Value-Matched Ensemble Pairs

The right panel of Figure 1 illustrates how the left panel simulation could be applied to create pairs of value-matched ensembles that differ in component evaluation certainty. It uses the same clustering approach that we use in the experiments (see *Details about the Clustering Algorithm* in these SOM). We then thin out the ensembles for plotting purposes, selecting the first ensemble in each 0.05 interval of the evaluation scale.

Experiment 1: K-fold cross validation

To compare the capacity of different predictors in estimating the actual evaluation certainty of ensembles, we employed K-fold Cross-Validation to assess model performance on left-out data. In K-fold Cross-Validation, the data set is divided into K segments or folds. For each fold, the model is trained on K-1 segments and tested on the remaining segment, thereby allowing us to evaluate the model's predictive accuracy on data it has not previously seen. This method provides a practical assessment of out-of-sample predictive capability,

akin to traditional model fit indices like AIC or WAIC, but through empirical evaluation rather than estimation.

In our specific application, we utilized a 10-fold setup, assigning 90% of the data to model training and the remaining 10% to testing in each iteration. We conducted two distinct types of validation: one leaving out two ensembles per participant (leave-out-ensemble) and another leaving out all data from approximately 10% of the participants (leave-out-participant). This dual approach allows us to gauge model robustness both when encountering new ensembles from known participants and when predicting outcomes for entirely new participants.

The results, detailed in Table S1, indicate varied model performances under different conditions:

- Models excluding overall value and value extremity predictors showed that the component-item certainty model consistently outperformed the single-item value similarity models across both leave-out scenarios.
- When including item-value covariates, the certainty models maintained superior performance, though the distinction from the value similarity model became less pronounced in the leave-out-ensemble fold, and no clear performance advantage was observed for the zero-one-inflated beta models.
- Across all configurations, component evaluation certainty models surpassed the covariate-only models, reinforcing the importance of component-item certainty in predicting ensemble evaluation certainty.

These findings underscore the utility of component-item certainty as a robust predictor in the context of ensemble evaluation.

Experiment 1: Ensemble Matching Manipulation Checks

The sorting and categorization algorithm performed well in creating the desired ensemble properties. Specifically, the low component evaluation certainty ensembles (category 1 and 2) scored consistently lower on average component evaluation certainty

Table 1
Results of K-fold Cross Validation

model	k-fold elpd	SD	Δ elpd	Δ SD	Δ elpd / Δ SD
leave-out-subject (no covariates)					
component evaluation certainty	-16601.60	74.03	-	-	-
value similarity	-17050.79	74.50	449.19	41.75	10.76
leave-out-ensemble (no covariates)					
component evaluation certainty	-15935.00	94.72	-	-	-
value similarity	-16572.42	96.50	637.41	80.17	7.95
leave-out-subject					
component evaluation certainty	-16564.46	79.05	-	-	-
value similarity	-16598.15	78.96	33.69	9.38	3.59
value-based covariates	-16634.75	80.67	70.29	12.65	5.56
leave-out-ensemble					
component evaluation certainty	-15606.78	102.61	-	-	-
value similarity	-15658.03	108.09	51.25	27.23	1.88
value-based covariates	-15716.76	106.81	109.98	27.21	4.04

compared to their counterparts (category 3 and 4; Estimate = 33.36, 95%CI = [29.10, 37.79], $pp_- < .001$) with only small, though credible, variations across evaluation similarity ensembles (categories 1 and 3 vs. 2 and 4) (Estimate = 1.45, 95%CI = [0.50, 2.42], $pp_- = .002$) Similarly, the low evaluation similarity ensembles (category 1 and 3) scored lower on evaluation similarity compared to their counterparts (category 2 and 4) (Estimate = 20.61, 95%CI = [18.70, 22.58], $pp_- < .001$). Again, evaluation similarity differed only slightly between the ensembles created to differ on component evaluation certainty (Estimate = 5.34, 95%CI = [1.47, 9.10], $pp_- = .002$). Altogether, this suggests that the ensemble types varied across the expected dimension in the expected direction without meaningful variations on the other dimension.

Experiment 1: Skew of evaluation certainty distribution

As Figure S1 shows, the distribution of ensemble-evaluation certainty judgments was heavily left-skewed, indicating that most of the certainty judgements were very high, with a mean of 156.11 and a median of 163.00.

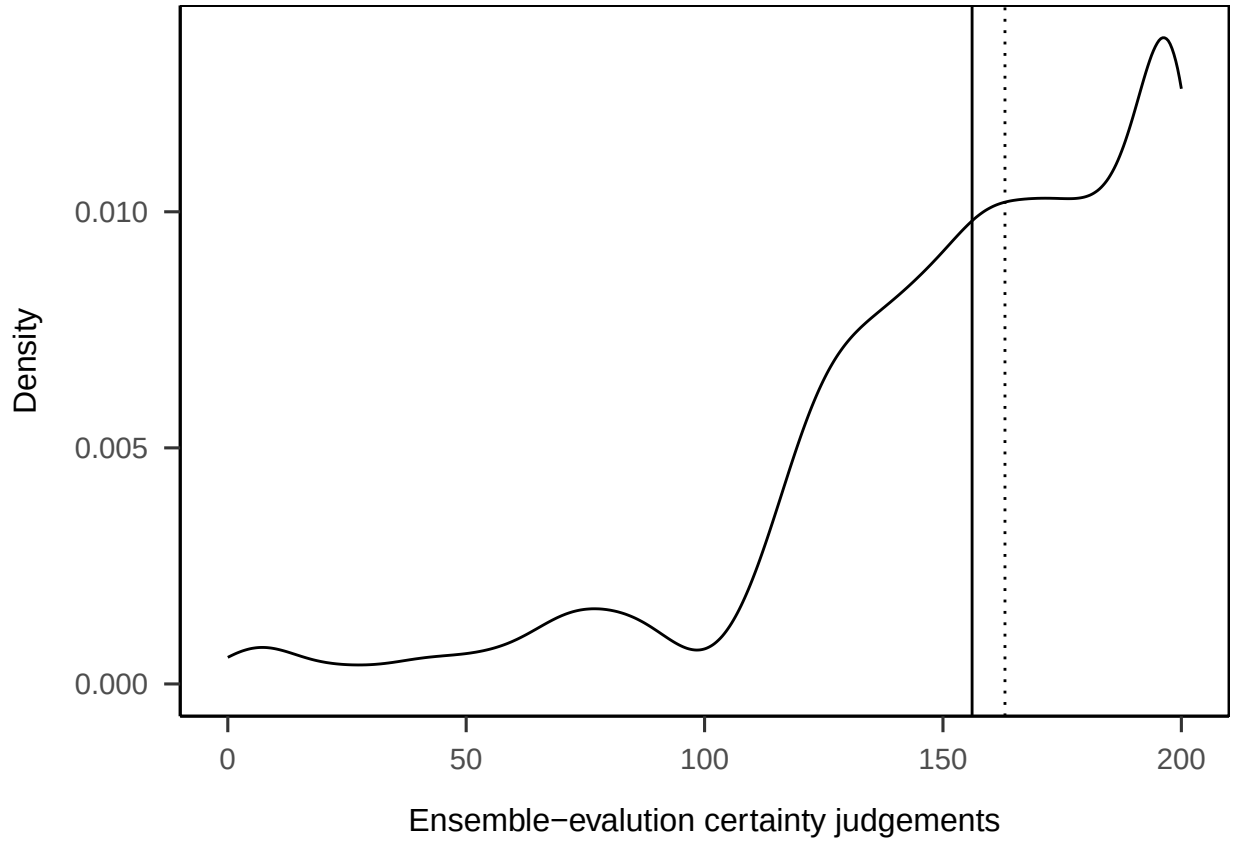

Figure 1
Density plot of ensemble-evaluation certainty judgments. Solid line displays distribution mean, and dotted line displays distribution median.

Deviations from preregistration

Experiment 1

We stated that that we would use numeric predictors for evaluation certainty and evaluation similarity (quantified as the SD between items in an ensemble), but changed this in the manuscript to the categorical predictors to increase consistency with the analyses reported for the other Experiments. The original results did however, provide identical conclusions:

Component evaluation certainty (Estimate = 0.20, 95%CI = [0.15, 0.25], $pp_- < .001$), and evaluation similarity (Estimate = -0.11, 95%CI = [-0.14, -0.08], $pp_+ < .001$) were credible predictors of the ensemble evaluation certainty. When comparing the size of the estimates component evaluation certainty (Estimate = 0.20, 95%CI = [0.15, 0.25], $pp_- <$

.001) was a stronger predictor than evaluation similarity (Estimate = -0.11, 95%CI = [-0.14, -0.08], $pp_+ < .001$), with a credible difference between the estimate sizes (Estimate = 0.09, 95%CI = [0.04, 0.13], $pp_- = .001$). This numerical difference was substantiated by K-fold cross validation showing that component evaluation certainty consistently outperformed evaluation similarity at predicting held-out ensemble evaluation certainty (see Cross Validation section and Table S1 in these Supplemental Materials).

Experiment 2

We did not preregister including a random intercept for ensembles in the model. However, such an intercept makes sense as Experiment 2 includes 3 evaluations and certainty ratings per ensemble, which are likely to be correlated within ensembles. The inclusion of this random intercept did not change the results of the model, and the conclusions remained the same. Component evaluation certainty condition, Estimate = 0.19, 95%CI = [0.12, 0.27], $pp_- < .001$, value positivity: Estimate = 0.18, 95%CI = [0.03, 0.32], $pp_- = .011$, value similarity: Estimate = -0.15, 95%CI = [-0.23, -0.08], $pp_+ < .001$ and value extremity: Estimate = 0.15, 95%CI = [0.09, 0.22], $pp_- < .001$ were credible predictors of ensemble evaluation certainty.

Model family comparisons

We defined four models in the preregistrations of Experiment 1 to 3 that we considered as candidates to describe the data-generating process: a Gaussian model, a Skew-Normal model, a Beta-Binomial model and a Zero-One-Inflated Beta (ZOIB) model, all of which use the default priors that are implemented in the brms package. The models were again compared and the Beta-Binomial model is reported in the main paper (and the only one used in Experiment 4) due to better performance on recovering data parameters and fit to the response distribution.

Experiments 2 to 4: Power analysis

Using the posterior predictions for possible effect sizes based on the respective previous Experiments (i.e., the data of Experiment 1 for Experiment 2 and the data of

Table 2

Model parameter values for predictors in Experiment 1 to 3 (pp equivalence indicates posterior proportion opposite to the prediction falling inside the ROPE)

Predictor	estimate	95% CI	pp	pp equivalence
Experiment 1 (gaussian)				
average CEC	0.07	0.05 0.09	< .001	-
value similarity	0.05	0.03 0.06	< .001	-
Experiment 1 (ZOIB)				
average CEC	0.14	0.10 0.17	< .001	-
value similarity	0.09	0.07 0.12	< .001	-
Experiment 2 (gaussian)				
item certainty condition	0.02	0.01 0.03	< .001	.947
value positivity	0.03	0.02 0.04	< .001	.139
value similarity	0.02	0.02 0.03	< .001	.647
Experiment 2 (ZOIB)				
component certainty condition	0.02	0.01 0.04	.005	.600
value positivity	0.06	0.03 0.08	< .001	.957
value similarity	0.01	-0.01 0.03	.119	.957
Experiment 3 (gaussian)				
component certainty condition	0.03	0.02 0.04	< .001	.233
value positivity	0.03	0.02 0.05	< .001	.120
value similarity	0.01	0 0.02	.006	.991
value extremity	0.03	0.02 0.04	< .001	.237
matching variable	-0.00	-0.01 0.01	.622	1.000
component certainty condition X matching variable	-0.00	-0.01 0.00	.898	1.000
Experiment 3 (ZOIB)				
component certainty condition	0.05	0.03 0.06	< .001	.004
value positivity	0.08	0.05 0.11	< .001	.001
value similarity	0.03	0.01 0.04	.001	.490
value extremity	0.06	0.04 0.08	< .001	.002
matching variable	0.00	-0.01 0.02	.247	.999
component certainty condition X matching variable	-0.01	-0.02 0.00	.968	1.000

Experiment 2 for Experiment 3 and 4), we conducted a power simulation of 4 steps. First, a multivariate zero-one-inflated beta model model was fit on the evaluations and certainty judgements of the individual items. The multivariate zero-one-inflated beta (ZOIB) model is a specialized finite mixture-model designed to effectively handle data characterized by a preponderance of boundary values (0 and 1) and continuous variation within the open interval (0, 1). This model categorizes responses into two distinct processes: boundary responses that are precisely at the extremes (0 or 1) and intermediate responses that occur between these extremes. Employing the ZOIB model allows for separate estimation of these two types of responses, thereby facilitating more accurate statistical analysis in datasets where extreme values are disproportionately represented. This approach is particularly advantageous for simulations, where the data-generating process is known to produce a high number of boundary responses. Second, from this model, posterior predictions were created for item values and certainties that served as simulated evaluations and certainty judgement for possible future observations. Third, the clustering algorithm was applied to each simulated data set, creating simulated ensembles that were matched on value and differed on certainty. Finally, the simulated clustered ensembles were used to fit the outcome model that would predict the ensemble certainty based on the simulated ensembles' component certainty and value similarity. Thus, this simulation strategy ensured realistic simulations of ensemble properties and ensemble counts for each participant, based on the actually observed data in the previous experiments.

Experiments 2 to 4: Details about the clustering algorithm

The clustering algorithm used euclidean distance with an average linkage to create k clusters where k was the number of rated items for a given participant divided by 6 to create at least 2 ensembles that contained items of similar value per cluster. The exact number of ensembles that the algorithm created per participant depended on the individual ratings of each participant. This set of ensembles would contain pairs in which 2 ensembles would be of approximately equal value as they contain items from the same cluster. Hence, the algorithm

would create between 2 and 10 ensembles per participant, as with 60 items, a maximum of 10 ensemble pairs could be created. Within each cluster, in a second step, the items were ranked based on their certainty, and after excluding the highest and the lowest certainty item from each cluster in order to exclude potential outliers, the 3 highest-certainty items formed a *high-certainty* ensemble of the cluster and the 3 lowest-certainty items formed a *low-certainty* ensemble of the cluster.

Experiments 2 to 4: Details on Data Analysis

In Experiment 2 and 3, we estimated if the 95% credible interval of the posterior difference between ensemble types was in between -1.25% and 1.25% of the entire scale width. If this was the case, and the 95% credible interval would include 0, we would consider the effect precisely estimated and that it did not, in an important way, impact ensemble evaluation certainty. While the choice of the ROPE size was arbitrary, we included this ROPE to ensure that we would not over-interpret small effects when making claims about persistent effects of component certainty on ensemble certainty. If the 95% credible interval partly falls within the ROPE, but does not include 0, we consider the effect credible but too small to be a major contributor to ensemble evaluation certainty. If the 95% credible interval falls entirely outside the ROPE, we consider the effect a major contributor to ensemble evaluation certainty. Note that there is a mistake in the preregistration, incorrectly stating that the ROPE corresponds to a difference between zero-sum coded groups of 5% of the scale. However, given the [-0.0125, 0.0125] ROPE this should be 2.5%, not 5%, as zero-sum coded estimates for two groups represent the group difference divided by two. For example, an estimate of 0.0125 corresponds to a group difference of 0.025 which is 2.5 percent of the scale.

Experiments 2 to 4: Ensemble Matching Manipulation Checks

Glossary of Statistical Terms

- **Posterior Proportion (pp)**: Compared to a frequentist model, a Bayesian model does not only provide point estimates for effect sizes, but a full distribution of possible effect sizes, given the prior and the observed data. Here, the posterior proportion

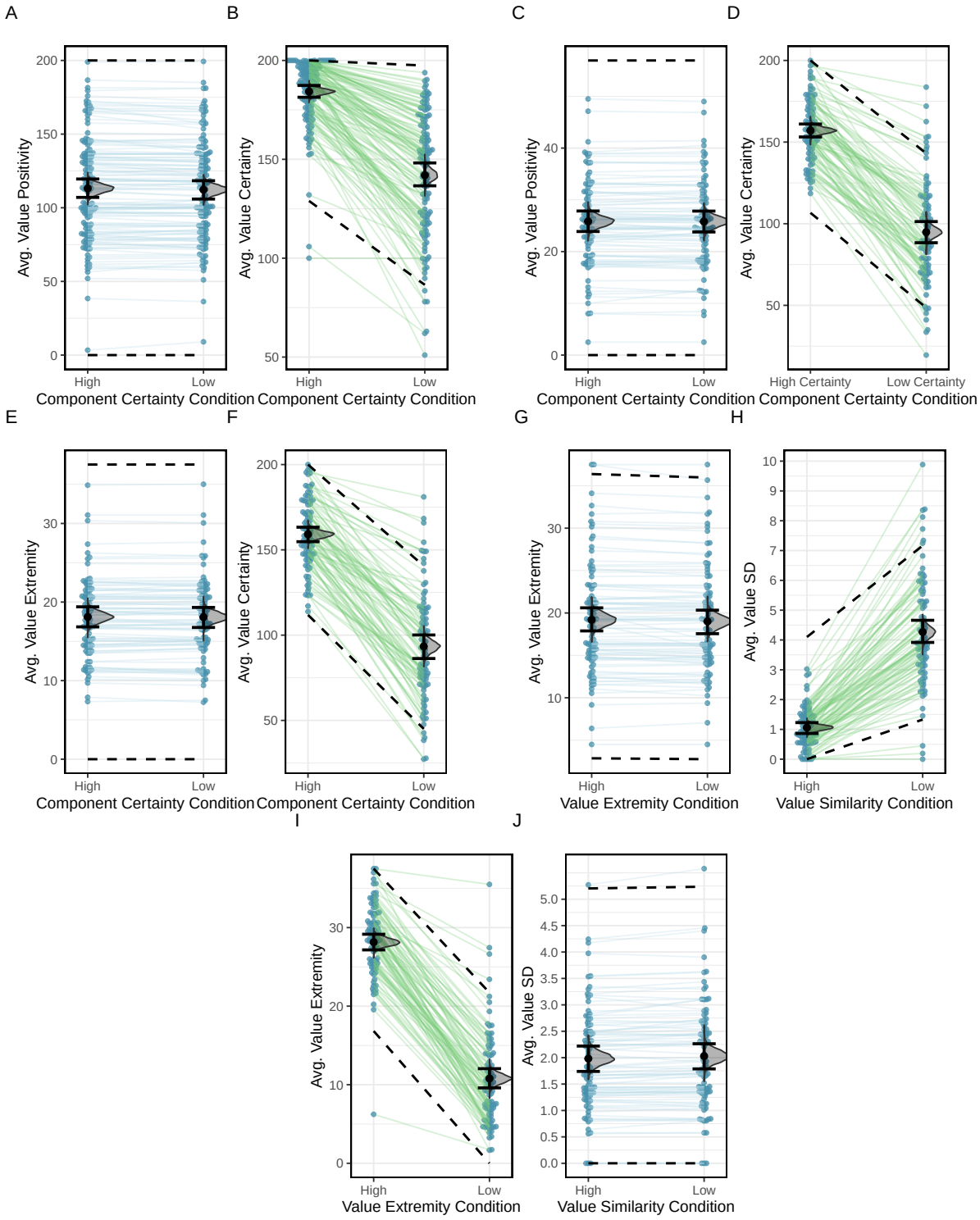

Figure 2

Overview of Basket-Matching Success in Experiments 2 to 4. Blue dots in each plot represent participant-wise average values. Green lines indicate participant-wise averages in line with plotted prediction, red contrary to prediction. Blue lines are used in plots where we expected no difference (i.e. for the variables that were matched). A: Experiment 2, average value positivity across positivity-matched baskets. B: Experiment 2, average component evaluation certainty across positivity-matched ensembles. C: Experiment 3, average value positivity across positivity-matched baskets. D: Experiment 3, average component evaluation certainty across positivity-matched ensembles. E: Experiment 3, average value extremity across extremity-matched baskets. F: Experiment 3, average component evaluation certainty across extremity-matched ensembles. G: Experiment 4, average value extremity across extremity-matched baskets. H: Experiment 4, average value SD across extremity-matched ensembles. I: Experiment 4, average value extremity across similarity-matched baskets. J: Experiment 4, average value SD across similarity-matched ensembles.

denoted as *pp* is the proportion of the posterior distribution that falls in the opposite direction of the hypothesis, either above or below zero or above or below the ROPE (see below). For example, if the hypothesis is that the effect is positive, a *pp* of 0.05 would indicate that 5% of the posterior distribution falls in the negative direction. While it might at first seem more intuitive to report the proportion of the posterior distribution that falls in the hypothesized direction, the *pp* is more informative as it indicates the proportion of the posterior distribution that is in conflict with the hypothesis, and is therefore somewhat more closer in interpretation to p-values, though the two do not share the same interpretation. Yet, to further facilitate interpretation, we stick to conventional frequentist norms in the field of considering an effect credible when less than 2.5% of the posterior distribution falls in the opposite direction of the hypothesis (i.e. $pp < 0.025$), to align with the conventional alpha level of 0.05 (i.e., one-sided 0.025) in frequentist statistics.

- **Prior** is a distribution that represents the researcher's beliefs about the effect size before observing the data. In Bayesian statistics, the prior is combined with the likelihood of the data to obtain the posterior distribution, which represents the researcher's beliefs about the effect size after observing the data. The prior can be informative or uninformative. An informative prior is a prior that is based on previous research or expert knowledge about the effect size. An uninformative prior is a prior that is based on little or no information about the effect size. In this study, we used uninformative priors for all analyses. While uninformative priors do not convey a lot of information and do not influence the posterior distribution much, they are useful for estimating complex models, as they can help to regularize the estimates and prevent overfitting.
- **credible** is a term used in Bayesian statistics that denotes whether an effect is considered to be credible given the data and the prior. Here, an effect is considered credible if less than 2.5% of the posterior distribution falls in the opposite direction of

the hypothesis. Bayesian estimates are usually accompanied by a credible interval or a Highest Posterior Density Interval (HPDI) that denotes a specific range of values (e.g. 95%), in which the true effect is likely to fall given the prior and observed data.

- **Region of Practical Equivalence (ROPE)** is an approach that is comparable to equivalence testing in frequentist statistics. It is used to determine if an effect is practically equivalent to zero. In Bayesian statistics, the ROPE is defined as a range of values that are considered to be practically equivalent to zero. If the 95% credible interval of an effect falls entirely within the ROPE, the effect is considered to be practically equivalent to zero. If the 95% credible interval falls entirely outside the ROPE, the effect is considered to be practically different from zero. If the 95% credible interval partly falls within the ROPE, but does not include zero, the effect is considered to be credible but too small to be a major contributor to the outcome.

Specifications of statistical models reported in the main text

Experiment 1

- Main hypothesis test:

```
Family:beta_binomial
Links:mu = logit
phi = identity
Formula:ensemble_certainty | trials(200) ~ value_similarity_f *
    value_certainty_f + (1 + value_similarity_f * value_certainty_f |
    participant)
```

Experiment 2

- Main hypothesis test:

```
Family:beta_binomial
Links:mu = logit
phi = identity
Formula:ensemble_certainty | trials(200) ~ value_certainty_f +
```

```
    value_positivity_s + value_similarity_s + value_extremity_s +
    (1 + value_certainty_f + value_positivity_s + value_similarity_s +
        value_extremity_s | participant) + (1 | ensemble_id)
```

- Within value-matched pair model:

```
Family:student
Links:mu = identity
sigma = identity
nu = identity
Formula:ensemble_certainty_difference ~ value_extremity_s + value_certainty_f +
    value_similarity_s + (1 + value_extremity_s + value_certainty_f +
    value_similarity_s | participant)
```

Experiment 3

- Main hypothesis test (separately fit for ensembles matched on value positivity and extremity):

```
Family:beta_binomial
Links:mu = logit
phi = identity
Formula:ensemble_certainty | trials(200) ~ value_certainty_f +
    value_positivity_s + value_similarity_s + value_extremity_s +
    (1 + value_certainty_f + value_positivity_s + value_similarity_s +
        value_extremity_s | participant)
```

- Within value-matched pair model (positivity-matched ensembles)

```
Family:student
Links:mu = identity
sigma = identity
nu = identity
Formula:ensemble_certainty_difference ~ value_extremity_s + value_certainty_f +
```

```
    value_similarity_s + (1 + value_extremity_s + value_certainty_f +
    value_similarity_s | participant)
```

- Within value-matched pair model (extremity-matched ensembles)

```
Family:student
Links:mu = identity
sigma = identity
nu = identity
Formula:ensemble_certainty_difference ~ value_positivity_s +
    value_certainty_f + value_similarity_s + (1 + value_positivity_s +
    value_certainty_f + value_similarity_s | participant)
```

- Model predicting correspondence between component and ensemble evaluations:

```
Family:lognormal
Links:mu = identity
sigma = identity
Formula:component_vs_ensemble_evaluation_difference ~ value_certainty_f *
    experiment_indicator + (1 + value_certainty_f | participant) +
    (1 | ensemble_id)
```

Experiment 4

- Main hypothesis test:

```
Family:beta_binomial
Links:mu = logit
phi = identity
Formula:ensemble_certainty | trials(200) ~ value_certainty_f *
    ensemble_matching_variable_f + (1 + value_certainty_f * ensemble_matching_variable_f |
    participant)
```

The above model formulas are in brms / lme4 syntax. The | symbol indicates that the right-hand side of the formula is a grouping factor. The * symbol indicates that the

right-hand side of the formula is an interaction term. The `+` symbol indicates that the right-hand side of the formula is an additive term. The `_s` suffix indicates that the variable is standardized. The `_f` suffix indicates that the variable is a sum-to-zero coded factorial variable.

Software overview

We used the following software during programming, analyses and writing the paper:

We used R version 4.5.0 (R Core Team, 2025) and the following R packages: afex v. 1.4.1 (Singmann et al., 2024), brms v. 2.22.12 (Bürkner, 2017, 2018, 2021), cmdstanr v. 0.9.0 (Gabry et al., 2025), cowplot v. 1.1.3 (Wilke, 2024), devtools v. 2.4.5 (Wickham et al., 2022), doParallel v. 1.0.17 (Corporation & Weston, 2022), doRNG v. 1.8.6.2 (Gaujoux, 2025), emmeans v. 1.11.1 (Lenth, 2025), flextable v. 0.9.8 (Gohel & Skintzos, 2025), foreach v. 1.5.2 (Microsoft & Weston, 2022), future v. 1.49.0 (Bengtsson, 2021), ggbeeswarm v. 0.7.2 (Clarke et al., 2023), gridExtra v. 2.3 (Auguie, 2017), HDInterval v. 0.2.4 (Meredith & Kruschke, 2022), here v. 1.0.1 (Müller, 2020), iterators v. 1.0.14 (Analytics & Weston, 2022), listenv v. 0.9.1 (Bengtsson, 2024), lme4 v. 1.1.37 (Bates et al., 2015), loo v. 2.8.0.9000 (Vehtari et al., 2017, 2024; Yao et al., 2018), Matrix v. 1.7.3 (Bates et al., 2025), officer v. 0.6.8 (Gohel et al., 2025), osfr v. 0.2.9 (Wolen et al., 2020), papaja v. 0.1.3 (Aust & Barth, 2024), patchwork v. 1.3.0 (Pedersen, 2024), plyr v. 1.8.9 (Wickham, 2011), prereg v. 0.6.0 (Aust & Spitzer, 2022), Rcpp v. 1.0.14 (Eddelbuettel, 2013; Eddelbuettel et al., 2025; Eddelbuettel & Balamuta, 2018; Eddelbuettel & François, 2011), rngtools v. 1.5.2 (Gaujoux, 2021), rstan v. 2.36.0.9000 (Stan Development Team, n.d.), StanHeaders v. 2.36.0.9000 (Stan Development Team, 2020), tidyverse v. 2.0.0 (Wickham et al., 2019), tinylabels v. 0.2.5 (Barth, 2025), usethis v. 3.1.0 (Wickham et al., 2024).

References

Analytics, R., & Weston, S. (2022). *iterators: Provides iterator construct*. https://doi.org/10.32614/CRAN.package.iterators

Auguie, B. (2017). *gridExtra: Miscellaneous functions for "Grid" graphics*.

https://doi.org/10.32614/CRAN.package.gridExtra

Aust, F., & Barth, M. (2024). *papaja: Prepare reproducible APA journal articles with R Markdown*. https://doi.org/10.32614/CRAN.package.papaja

Aust, F., & Spitzer, L. (2022). *prereg: R markdown templates to preregister scientific studies*. https://doi.org/10.32614/CRAN.package.prereg

Barth, M. (2025). *tinylabels: Lightweight variable labels*. https://doi.org/10.32614/CRAN.package.tinylabels

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bates, D., Maechler, M., & Jagan, M. (2025). *Matrix: Sparse and dense matrix classes and methods*. https://doi.org/10.32614/CRAN.package.Matrix

Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in r using futures. *The R Journal*, *13*(2), 208–227. https://doi.org/10.32614/RJ-2021-048

Bengtsson, H. (2024). *listenv: Environments behaving (almost) as lists*. https://doi.org/10.32614/CRAN.package.listenv

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411. https://doi.org/10.32614/RJ-2018-017

Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, *100*(5), 1–54. https://doi.org/10.18637/jss.v100.i05

Clarke, E., Sherrill-Mix, S., & Dawson, C. (2023). *ggbeeswarm: Categorical scatter (violin point) plots*. https://doi.org/10.32614/CRAN.package.ggbeeswarm

Corporation, M., & Weston, S. (2022). *doParallel: Foreach parallel adaptor for the "parallel" package*. https://doi.org/10.32614/CRAN.package.doParallel

Eddelbuettel, D. (2013). *Seamless R and C++ integration with Rcpp*. Springer.

https://doi.org/10.1007/978-1-4614-6868-4

Eddelbuettel, D., & Balamuta, J. J. (2018). Extending R with C++: A Brief Introduction to Rcpp. *The American Statistician*, *72*(1), 28–36. https://doi.org/10.1080/00031305.2017.1375990

Eddelbuettel, D., Francois, R., Allaire, J., Ushey, K., Kou, Q., Russell, N., Ucar, I., Bates, D., & Chambers, J. (2025). *Rcpp: Seamless r and c++ integration*. https://doi.org/10.32614/CRAN.package.Rcpp

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, *40*(8), 1–18. https://doi.org/10.18637/jss.v040.i08

Gabry, J., Češnovar, R., Johnson, A., & Bronder, S. (2025). *cmdstanr: R interface to "CmdStan"*. https://mc-stan.org/cmdstanr/

Gaujoux, R. (2021). *rngtools: Utility functions for working with random number generators*. https://doi.org/10.32614/CRAN.package.rngtools

Gaujoux, R. (2025). *doRNG: Generic reproducible parallel backend for "foreach" loops*. https://doi.org/10.32614/CRAN.package.doRNG

Gohel, D., Moog, S., & Heckmann, M. (2025). *officer: Manipulation of microsoft word and PowerPoint documents*. https://doi.org/10.32614/CRAN.package.officer

Gohel, D., & Skintzos, P. (2025). *flextable: Functions for tabular reporting*. https://doi.org/10.32614/CRAN.package.flextable

Lenth, R. V. (2025). *emmeans: Estimated marginal means, aka least-squares means*. https://doi.org/10.32614/CRAN.package.emmeans

Meredith, M., & Kruschke, J. (2022). *HDInterval: Highest (posterior) density intervals*. https://doi.org/10.32614/CRAN.package.HDInterval

Microsoft, & Weston, S. (2022). *foreach: Provides foreach looping construct*. https://doi.org/10.32614/CRAN.package.foreach

Müller, K. (2020). *here: A simpler way to find your files*. https://doi.org/10.32614/CRAN.package.here

Pedersen, T. L. (2024). *patchwork: The composer of plots*. https://doi.org/10.32614/CRAN.package.patchwork

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Singmann, H., Bolker, B., Westfall, J., Aust, F., & Ben-Shachar, M. S. (2024). *afex: Analysis of factorial experiments*. https://doi.org/10.32614/CRAN.package.afex

Stan Development Team. (n.d.). *RStan: The R interface to Stan*. https://mc-stan.org/

Stan Development Team. (2020). *StanHeaders: Headers for the R interface to Stan*. https://mc-stan.org/

Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T., & Gelman, A. (2024). *loo: Efficient leave-one-out cross-validation and WAIC for bayesian models*. https://mc-stan.org/loo/

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*, 1413–1432. https://doi.org/10.1007/s11222-016-9696-4

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, *40*(1), 1–29. https://www.jstatsoft.org/v40/i01/

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., Bryan, J., Barrett, M., & Teucher, A. (2024). *usethis: Automate package and project setup*. https://doi.org/10.32614/CRAN.package.usethis

Wickham, H., Hester, J., Chang, W., & Bryan, J. (2022). *devtools: Tools to make developing r packages easier*. https://doi.org/10.32614/CRAN.package.devtools

Wilke, C. O. (2024). *cowplot: Streamlined plot theme and plot annotations for "ggplot2"*.

https://doi.org/10.32614/CRAN.package.cowplot

Wolen, A. R., Hartgerink, C. H. J., Hafen, R., Richards, B. G., Soderberg, C. K., & York, T. P. (2020). osfr: An R interface to the open science framework. *Journal of Open Source Software*, *5*(46), 2071. https://doi.org/10.21105/joss.02071

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Using stacking to average bayesian predictive distributions. *Bayesian Analysis*, *13*, 917–1007. https://doi.org/10.1214/17-BA1091