Distinguishing representational geometries with controversial stimuli: Bayesian experimental design and its application to face dissimilarity judgments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Comparing representations of complex stimuli in neural network layers to human brain representations or behavioral judgments can guide model development. However, even qualitatively distinct neural network models often predict similar representational geometries of typical stimulus sets. We propose a Bayesian experimental design approach to synthesizing stimulus sets for adjudicating among representational models efficiently. We apply our method to discriminate among candidate neural network models of behavioral face similarity judgments. Our results indicate that a neural network trained to invert a 3D-face-model graphics renderer is more human-aligned than the same architecture trained on identification, classification, or autoencoding. Our proposed stimulus synthesis objective is generally applicable to designing experiments to be analyzed by representational similarity analysis for model comparison.

Neural network models have become increasingly important in cognitive computational neuroscience to avoid the ambiguity of verbal hypotheses and quantitatively capture behavior and neural activity. As our representational models improve, we encounter a new challenge. Models of human perceptual representation are typically evaluated with respect to natural or naturalistic stimulus sets (e.g., photographs or rendered images of 3D objects) chosen by the experimenter. However, there is no guarantee that the stimulus set we choose will reveal the distinct representational geometries in the models we wish to compare. Several recent studies find that data sets are explained equally well by multiple qualitatively distinct models. For example, in a study employing natural images as stimuli, multiple neural network architectures showed similar level of correspondence with human neural representations [1]. The challenge of similar representational geometries predicted by distinct models is even more pronounced in the subdomain of face representation, where natural face stimuli span a relatively limited image manifold. Jozwik et al. [2] found that several qualitatively distinct computational models predicted human behavioral face dissimilarity judgments almost perfectly. Equivalent accuracy levels of distinct face representation models were also observed when the model dissimilarities were compared to neural representational dissimilarities measured by fMRI [3] or ECoG [4]. While the above studies used representational similarity analysis [RSA 5] to compare human and model representations, the problem of non-discriminable predictions of distinct models is not unique to RSA. Model discrimination is expected to be even more challenging in encoding analyses, where the many parameters of the fitted linear mapping render each model more flexible and reduce model discriminability even in ideal, noiseless conditions [6, 7].

One way forward is to test model-brain alignment with controversial stimuli: stimuli generated to induce distinct predictions in different models [8–10]. These studies optimized the controversiality of the stimuli in terms of the image labels or relative sentence probabilities assigned by different

Submitted to 4th Workshop on Shared Visual Representations in Human and Machine Visual Intelligence (SVRHM) at NeurIPS 2022. Do not distribute.

models, reflecting the human behavioral tasks used to adjudicate among the models. Here, we put forward a stimulus synthesis method that generates stimulus sets for which different models predict distinct *representational geometries*. Our proposed method can compare fixed models by their unweighted representational geometries (i.e., "fixed RSA", see [11]) or flexible models whose weighting parameters have already been estimated. We apply the proposed method to discriminate among a set of neural network models of behavioral judgments of face similarity. The proposed optimization objective for stimulus synthesis is applicable for designing RSA experiments in general, beyond the domain of faces, and for experiments that measure either brain activity or behavioral judgments.

1 Methods

1.1 Stimulus optimization objective

In order to generate a stimulus set that supports efficient model discrimination, we take a Bayesian Optimal Experimental Design (BOED) approach [12, 13]. Unlike typical BOED applications where the "design" consists of experimental protocol parameters such as stimulus presentation order and timing, here we optimize the experimental stimulus set itself.[1]

Differentiable model discrimination objective. To generate a stimulus set that efficiently discriminates between M representational models, we propose to maximize the following utility function:

$$U(\xi) = \sum_m p(m) f\Big(\underbrace{\psi(\tilde{\mathbf{y}}^m, \hat{\mathbf{y}}^m \mid \xi)}_{\substack{\text{how well model } m \text{ predicts} \\ \text{data generated by a held-out} \\ \text{instance of itself}}} - \underbrace{\max_{m' \neq m} \psi(\tilde{\mathbf{y}}^m, \hat{\mathbf{y}}^{m'} \mid \xi)}_{\substack{\text{how well the best other model} \\ \text{predicts data generated by} \\ \text{model } m}} \Big), \tag{1}$$

where $p(m)$ is our prior belief in model m (a uniform distribution if we have not collected any data yet), $\tilde{\mathbf{y}}^m$ is the vector of dissimilarities predicted by model m for stimulus set ξ, $\hat{\mathbf{y}}^m$ is the corresponding prediction vector made by another instance of model m with different weight initialization, ψ is a model-performance estimator (see below) and f is a monotonically increasing function, which we set here to emphasize negative differences (i.e., incorrect model recoveries) and de-emphasize positive differences (i.e., correct model recoveries). Here we used $f(x) = -e^{-10x}$, which led to high simulated model recovery accuracy compared to alternatives, including the identity function.

Model performance estimator. The model-performance estimator ψ should be as close as possible to how the human-model alignment will be evaluated in the analyses of the actual data, but needs to be differentiable to allow efficient stimulus optimization. In the case of neuroimaging RSA studies, each model is evaluated by its representational dissimilarity predictions across all stimulus pairs, yielding a representational dissimilarity matrix (RDM). Such RDMs can also be estimated from behavioral judgments [14]. To estimate model performance based on RDMs, whitened RDM correlation measures are appropriate [15]. For estimating model performance based on dissimilarities between independent stimulus pairs (where each stimulus appears only in one pair), simple correlation coefficients are usually adequate.

Accounting for multiple neural network layers. In studies that evaluate deep neural networks as computational models of human representational geometries, multiple layers of each neural network must be considered as potential representational models. A common approach is to evaluate each neural network by its most human-aligned layer [e.g., 16, 1, 17]. In principle, we can simply treat multiple layers as multiple models in the framework proposed above. However, such an approach might devote a large portion of the design's power to discriminating among the consecutive layers of single models (which tend to have related representations). Our main goal, however, is to adjudicate among entire neural network models. We therefore introduce layers as an unknown nuisance parameter in the experiment, yielding a slightly more elaborate optimization objective:

$$U(\xi) = \sum_m \sum_l p(m) p(l \mid m) f\Big(\max_{l'} \psi\big(\tilde{\mathbf{y}}^{m,l}(\xi), \hat{\mathbf{y}}^{m,l'}(\xi)\big) - \max_{m' \neq m} \max_{l'} \psi\big(\tilde{\mathbf{y}}^{m,l}(\xi), \hat{\mathbf{y}}^{m',l'}(\xi)\big)\Big), \tag{2}$$

where $p(l \mid m)$ is our prior belief that layer l is the data generating representation given model m, and $\tilde{\mathbf{y}}^{m,l}(\xi_t)$ and $\hat{\mathbf{y}}^{m,l}(\xi_t)$ are the vectors of dissimilarities predicted by this layer, according to two different instances. We refer the instances marked by hat (ˆ) as "reference" instances and reuse them in data analysis.

[1] We think of our controversial stimulus sets as "optimized" rather than "optimal" since we generate them by means of local optimization, which does not guarantee a globally optimal solution.

1.2 Optimizing face stimulus sets for discriminating among models of face dissimilarity judgments

We applied the approach described above to design controversial stimulus sets for a behavioral experiment that compares neural network models of face dissimilarity judgments.

We trained six neural networks sharing the veteran VGG-16 architecture [18] on different datasets and objectives (Table 1, Appendix A.1).

architecture	training task	training dataset
VGG-16	object categorization	ImageNet [19] (object photographs)
	face identification	VGGFace2 [20] (face photographs)
	face identification	BFM (synthetic faces)
	autoencoding (VAE)	VGGFace2 (face photographs)
	autoencoding (VAE)	BFM (synthetic faces)
	inverse rendering	BFM (synthetic faces)

Table 1: Six candidate neural network models of human face representation.

We used the 3D morphable Basel Face Model (BFM 2019; [21]) as our face stimulus generator. This generative model parameterizes individual face stimuli using separate latent spaces for shape, expression, and texture (see Appendix A.2). The latents define 3D face structure and texture independently from identity-invariant parameters such as pose and illumination.

We considered an experimental paradigm in which in every trial, participants arrange N face pairs along a dissimilarity axis [2] (Fig. S1). Model performance (i.e., how well model m predicts the response set y to stimulus set ξ) is quantified by first measuring the correlation between model dissimilarity predictions and the participant responses within each trial, and then averaging the resulting correlation coefficients across T trials:[2]

$$\psi(y, m \mid \xi) = \frac{1}{T} \sum_{t} \operatorname{corr}\left(\mathbf{y}_t, \hat{\mathbf{y}}^m(\xi_t)\right), \qquad (3)$$

where $\mathbf{y}_t$ is the vector of observed dissimilarity judgments in trial t and $\mathbf{y}^m(\xi_t)$ is the corresponding vector of dissimilarities predicted by model m. We used the squared Euclidean distance between the stimulus representations within each pair as the predicted dissimilarity. $\operatorname{corr}(\cdot, \cdot)$ is a correlation coefficient. We used Pearson's linear correlation coefficient for stimulus optimization and Spearman's rank correlation coefficient[3] for analyzing the human experiment (the former is differentiable, whereas the latter is invariant to order-preserving transformations). We constructed three multi-trial stimulus sets of BFM synthetic faces:

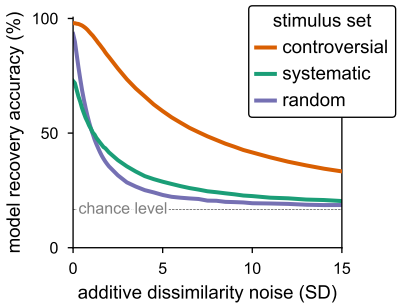

Figure 1: Simulated model recovery accuracy for three stimulus sets: (1) randomly sampled; (2) systematically sampled; and (3) controversial. One held-out instance per model was used to simulate observed dissimilarity vectors, assuming a uniform prior over the six models and 16 layers. The simulated dissimilarities were compared to the dissimilarities of the set of instances used as references in the stimulus optimization procedure. To simulate measurement noise, Gaussian noise was added to each representation (x-axis). The noise was scaled such that SD=1 would correspond to the SD of the representation's noiseless dissimilarities in the randomly sampled stimulus set. The y-axis denotes the percentage of simulations (n=2000) in which the data-generating model (regardless of the data-generating layer) was recovered.

Random stimulus set. As a baseline condition, we randomly sampled 144 faces from the BFM distribution (i.e., $\mathcal{N}(0, I)$) and rendered them as a frontal view. We randomly allocated the faces to pairs and assigned six face pairs to each of the 12 trials. Here, as well as in all other conditions, the BFM expression component was turned off, so variation was limited to the shape and texture components.

[2] This is a variation on [2], where the dissimilarities were first concatenated across trials and only then correlated with the model predictions, yielding a single correlation coefficient.

[3] Ties were handled by an analytical random-among-equals tie-breaking estimate [22, Eq. 14].

Systematically sampled stimulus set. For this stimulus set, we systematically sampled face pairs with different BFM distances. Following random-sampling and pairing, we constrained the BFM shape and texture latent vectors of each of 144 faces to have an l_{inf}-norm ≤ 2.0 (i.e., reside in a hyper-cube extending $\pm$ 2 SDs from the mean face in each dimension). We then adjusted the BFM shape and texture latents such that for each trial, one pair would have a zero Euclidean distance, one pair would have a maximal distance and occupy opposite corners of the hyper-cube, and the remaining four would evenly sample the range of distances in between (but not necessarily lie on the same line in face space). This was achieved by iterative constrained optimization, minimizing the squared deviations of the pairs' distances from the pre-determined distances.

Controversial stimulus set. Here, we used the same initialization and constraints but optimized model discriminability. We maximized Eq. 2 (plugging in Eq. 3) by iteratively adjusting the faces' latents while monitoring the representation of the resulting 2D face image in each layer of the six candidate models (evaluating two instances per model). We implemented a differentiable approximation to the BFM rendering process in PyTorch3D [23] to enable backpropagation of the gradients from the neural network representations through the 2D image to the BFM latents. This enabled us to use SGD optimization for stimulus synthesis (see further details in Appendix A.3).

1.3 Human behavioral experiment

We recruited 90 US-based, 18-60 year-old participants (39 female, 32.5$\pm$10.4 years mean age) through the `prolific.co` platform (Appendix A.4). Each stimulus set (random, systematic, and controversial) was tested on an independent set of 30 participants. The participants were presented with two repetitions of $T = 12$ unique trials in a randomized order. In each trial, the participants were instructed to arrange six face pairs according to their perceived dissimilarity along a vertical dissimilarity axis, labeled "identical" and "maximally different" on opposite ends (Figs. S1 and S2). The arrangement task was implemented on the Meadows web platform (`meadows-research.com`). Unlike the design in [2], no visual examples of identical and maximally different face pairs were provided. Each generated face image appeared only in one pair. Our experimental procedures were approved by the local institutional review board.

To obtain an unbiased estimate of each model's best layer performance, we evaluated each subject-model correlation using the model layer that best predicted the other 29 subjects' ratings (bottom panels in Fig. 2). A lower bound on the best possible model performance for each data set was calculated by predicting each participant's responses by the average response vector of the other 29 participants [24], calculated after rank-transforming the six ratings within each trial. An upper bound was calculated in the same fashion while including the predicted participant's response vector in the across-subject average.

2 Results

Simulations. We first considered the discriminability of the six models for each of the three stimulus sets in silico (Fig. 1). We simulated model recovery experiments by comparing the dissimilarities predicted by each model and each layer to the dissimilarities predicted by held-out model instances. This was done both in a noiseless setting and under Gaussian noise added to the simulated dissimilarities. As expected, the controversial stimulus set enabled more accurate model recovery, especially under noise. See Figure S3 for an ablation study of the optimization algorithm.

Behavioral results. Both randomly sampled and systematically sampled face pairs failed to successfully adjudicate among the six candidate models: no model was significantly more accurate than all of the others in predicting the human responses. In contrast, when models and humans were evaluated on the controversial stimulus set, the neural network trained on inverse rendering (i.e., predicting BFM latents from the input image) was found to be significantly more human-consistent than the five other models. We replicated this result with a second controversial stimulus set, optimized from a different random initialization of face latents (Fig. S4). Note that the systematically sampled stimulus set, which was designed to elicit maximally distinct levels of perceptual dissimilarity, indeed yielded highly reliable human judgments. Below the resulting high noise ceiling, however, the performance differences among the alternative models were small and largely insignificant. All models came close to the noise ceiling, illustrating how apparent good model performance can fail to drive theoretical progress.

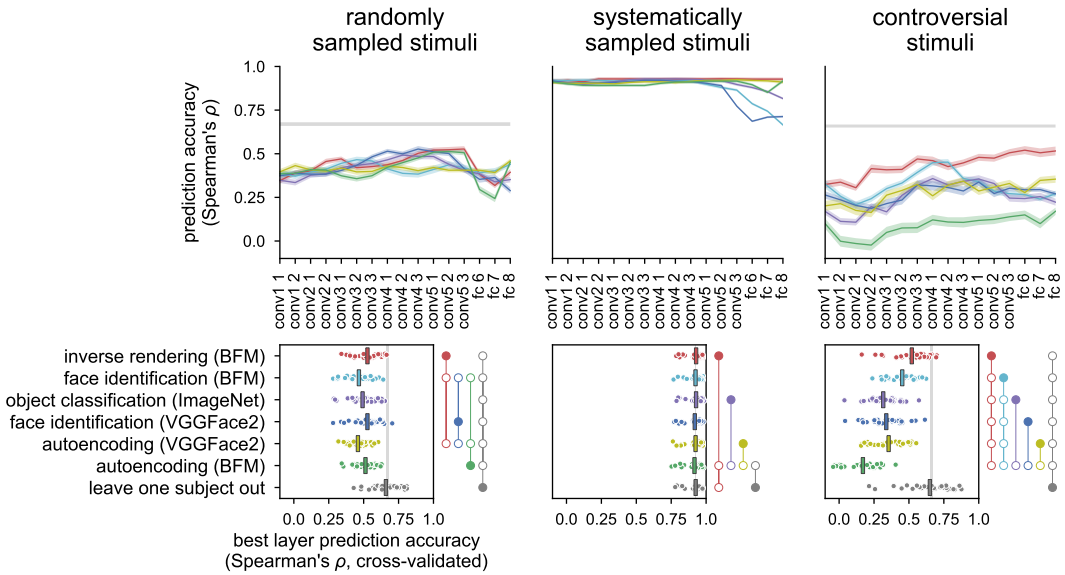

Figure 2: **Prediction accuracy of human similarity judgments of six VGG-16 neural networks trained on different tasks and datasets**. Each model is evaluated on a randomly sampled stimulus set (left column), a systematically sampled stimulus set (middle column), and a controversial stimulus set (right column). **Top row:** Each panel shows the average Spearman's ρ between the dissimilarity vectors of each model layer and human response for one stimulus set. Each colored line indicates performance of one model, and the shaded areas represent the SE. The grey regions indicate noise ceiling estimates on the average model-human alignment an ideal model can achieve, given the between-subject variability in human responses. **Bottom row:** Cross-validated performance estimates of each model's best layer. Each dot represents the rank correlation between one subject and one model, using the layer that best explained the other subjects' responses. Significance indicators: A solid dot connected to a set of open dots indicates that the model aligned with the solid dot has significantly higher correlation with the human judgments than any of the models aligned with the open dots ($p < .05$, paired t-tests on Fisher-transformed correlation coefficients, Holm-Šídák FWE-corrected for 21 comparisons).

3 Discussion

In this work, we put forward a controversial stimulus synthesis procedure for RSA experiments and applied it to compare models of face representation. Beyond faces, the proposed optimization approach can be readily applied to design controversial stimulus sets for differentiable representational models including neural networks, in vision and in other representational domains.

Limitations and future directions. Using gradient-based stimulus optimization enables efficient search and convergence in high-dimensional stimulus spaces (here, $\mathbb{R}^{398}$). However, gradient-based stimulus optimization requires differentiability of the objective, the stimulus renderer, and the models. Adapting evolutionary algorithm-based stimulus synthesis [25] to directly maximize model discrimination may allow a complementary approach (see [26] for an evolutionary approach to stimulus selection).

Implications for modeling human face representations. We found that a neural network trained to invert the graphics rendering process and estimate the BFM latent representations that gave rise to a face image yielded the most human-aligned representational geometry. This result is consistent with the finding of Yildirim et al. [27] obtained using macaque intracranial recordings [Cf., 28]. However, it should be noted that this network was tested in favorable conditions since it was trained on the generative model used to parameterize the experimental stimuli. In future work, we will evaluate the models' performance using controversial stimuli parameterized by alternative generators (e.g., generative adversarial networks) to address this limitation.

References

[1] Katherine R. Storrs, Tim C. Kietzmann, Alexander Walther, Johannes Mehrer, and Nikolaus Kriegeskorte. Diverse Deep Neural Networks All Predict Human Inferior Temporal Cortex Well, After Training and Fitting. *Journal of Cognitive Neuroscience*, 33(10):2044–2064, 09 2021. ISSN 0898-929X. doi:10.1162/jocn_a_01755.

[2] Kamila M. Jozwik, Jonathan O'Keeffe, Katherine R. Storrs, Wenxuan Guo, Tal Golan, and Nikolaus Kriegeskorte. Face dissimilarity judgments are predicted by representational distance in morphable and image-computable models. *Proceedings of the National Academy of Sciences*, 119(27):e2115047119, 2022. doi:10.1073/pnas.2115047119.

[3] Johan D. Carlin and Nikolaus Kriegeskorte. Adjudicating between face-coding models with individual-face fMRI responses. *PLOS Computational Biology*, 13(7):1–28, 2017. doi:10.1371/journal.pcbi.1005604.

[4] Shany Grossman, Guy Gaziv, Erin M. Yeagle, Michal Harel, Pierre Mégevand, David M. Groppe, Simon Khuvis, Jose L. Herrero, Michal Irani, Ashesh D. Mehta, and Rafael Malach. Convergent evolution of face spaces across human face-selective neuronal groups and deep convolutional networks. *Nature Communications*, 10(1):4934, 2019-12. ISSN 2041-1723. doi:10.1038/s41467-019-12623-6.

[5] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008. ISSN 1662-5137. doi:10.3389/neuro.06.004.2008.

[6] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019-06-09. URL `https://proceedings.mlr.press/v97/kornblith19a.html`.

[7] Yena Han, Tomaso Poggio, and Brian Cheung. System identification of neural systems: If we got it right, would we know?, 2022. URL `https://dspace.mit.edu/handle/1721.1/143617`.

[8] Tal Golan, Prashant C. Raju, and Nikolaus Kriegeskorte. Controversial stimuli: Pitting neural networks against each other as models of human cognition. *Proceedings of the National Academy of Sciences*, 117 (47):29330–29337, 2020-11-24. ISSN 0027-8424, 1091-6490. doi:10.1073/pnas.1912334117.

[9] Tal Golan and Nikolaus Kriegeskorte. Synthesizing Controversial Stimuli (a tutorial with PyTorch), November 2020. URL `https://github.com/kriegeskorte-lab/controversial_stimuli_tutorial`.

[10] Tal Golan, Matthew Siegelman, Nikolaus Kriegeskorte, and Christopher Baldassano. Testing the limits of natural language models for predicting human language judgments. *arXiv preprint arXiv:2204.03592*, 2022. doi:10.48550/arXiv.2204.03592.

[11] Seyed-Mahdi Khaligh-Razavi, Linda Henriksson, Kendrick Kay, and Nikolaus Kriegeskorte. Fixed versus mixed RSA: Explaining visual representations by fixed and mixed feature sets from shallow and deep computational models. *Journal of Mathematical Psychology*, 76:184 – 197, 2017. ISSN 0022-2496. doi:10.1016/j.jmp.2016.10.007.

[12] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10 (3):273–304, 1995. ISSN 08834237. doi:10.1214/ss/1177009939.

[13] Jay I. Myung and Mark A. Pitt. Optimal experimental design for model discrimination. *Psychological Review*, 116:499–518, 2009. ISSN 1939-1471(Electronic),0033-295X(Print). doi:10.1037/a0016104. Place: US Publisher: American Psychological Association.

[14] Nikolaus Kriegeskorte and Marieke Mur. Inverse MDS: Inferring dissimilarity structure from multiple item arrangements. *Frontiers in Psychology*, 3, 2012. ISSN 1664-1078. doi:10.3389/fpsyg.2012.00245.

[15] Jörn Diedrichsen, Eva Berlot, Marieke Mur, Heiko H. Schütt, Mahdiyar Shahbazi, and Nikolaus Kriegeskorte. Comparing representational geometries using whitened unbiased-distance-matrix similarity. *Neurons, Behavior, Data analysis, and Theory*, 5(3), 2021-08-23. ISSN 2690-2664. doi:10.51628/001c.27664.

[16] Martin Schrimpf, Jonas Kubilius, Michael J. Lee, N. Apurva Ratan Murty, Robert Ajemian, and James J. DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 108(3):413–423, 2020. ISSN 0896-6273. doi:10.1016/j.neuron.2020.07.040.

[17] Colin Conwell, Jacob S. Prince, George A. Alvarez, and Talia Konkle. Large-scale benchmarking of diverse artificial vision models in prediction of 7T human neuroimaging data. *bioRxiv*, 2022. doi:10.1101/2022.03.28.485868.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. doi:10.48550/arXiv.1409.1556.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi:10.1109/CVPR.2009.5206848.

[20] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74, Los Alamitos, CA, USA, may 2018. IEEE Computer Society. doi:10.1109/FG.2018.00020.

[21] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 75–82, 2018. doi:10.1109/fg.2018.00021.

[22] Heiko H. Schütt, Alexander D. Kipnis, Jörn Diedrichsen, and Nikolaus Kriegeskorte. Statistical inference on representational geometries. *arXiv preprint arXiv:2112.09200*, 2021-12-16. doi:10.48550/arXiv.2112.09200.

[23] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arxiv preprint arXiv:2007.08501*, 2020. doi:10.48550/arXiv.2007.08501.

[24] Hamed Nili, Cai Wingfield, Alexander Walther, Li Su, William Marslen-Wilson, and Nikolaus Kriegeskorte. A toolbox for representational similarity analysis. *PLOS Computational Biology*, 10(4):1–11, 2014. doi:10.1371/journal.pcbi.1003553.

[25] Carlos R. Ponce, Will Xiao, Peter F. Schade, Till S. Hartmann, Gabriel Kreiman, and Margaret S. Livingstone. Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, 177(4):999–1009.e10, 2019. ISSN 0092-8674. doi:10.1016/j.cell.2019.04.005.

[26] Kamila M Jozwik, Elias Najarro, Jasper JF Van Den Bosch, Ian Charest, Radoslaw M Cichy, and Nikolaus Kriegeskorte. Disentangling five dimensions of animacy in human brain and behaviour. *Communications Biology*, 2022. doi:10.17863/CAM.87987.

[27] Ilker Yildirim, Mario Belledonne, Winrich Freiwald, and Josh Tenenbaum. Efficient inverse graphics in biological face processing. *Science Advances*, 6(10):eaax5979, 2020. doi:10.1126/sciadv.aax5979.

[28] Christoph Daube, Tian Xu, Jiayu Zhan, Andrew Webb, Robin A.A. Ince, Oliver G.B. Garrod, and Philippe G. Schyns. Grounding deep neural network predictions of human categorization behavior in understandable functional features: The case of face identity. *Patterns*, 2(10):100348, 2021. ISSN 2666-3899. doi:10.1016/j.patter.2021.100348.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015. doi:10.1109/ICCV.2015.123.

[30] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020. doi:10.48550/arXiv.2009.00104.

[31] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. doi:10.3390/info11020125.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2014. doi:10.48550/arXiv.1312.6114.

[33] Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and effective VAE training with calibrated decoders. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9179–9189. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/rybkin21a.html`.

[34] Diederik P Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

[35] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pp. 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi:10.1145/311535.311556.

[36] Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. Individual differences among deep neural network models. *Nature communications*, 11(1):1–12, 2020. doi:10.1038/s41467-020-19632-w.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] The limitations are described in the discussion section.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not foresee negative societal impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We provided full instructions for reproducing the experimental results. We will share code and data in their entirety with the final, full-length report of this study.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Our code and models will be released with the final, full-length report of this study.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See Figure S1.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] There were no potential risks for the participants. Institutional Review Board information will appear in the non-anonymized version.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Appendix A.4.

A Appendix

A.1 Detailed description of model training

All of our models (Table 1) had the VGG16 architecture [18] as implemented in torchvision and were trained on 128 $\times$ 128 pixel input images. Neural network weights were initialized by randomly sampling from a zero-mean Gaussian distribution, as described in [29]. No batch normalization was employed. We used PyTorch Lightning for model training (`www.pytorchlightning.ai`). All the models were trained using four GeForce RTX 2080 Ti GPUs. We trained three instances for each of the six models described below, starting from different random weight initializations. The first instance was used as a representational model ($\hat{\mathbf{y}}^m$ in our equations). The second instance was used to simulate ground-truth responses during stimulus optimization ($\tilde{\mathbf{y}}^m$), and the third was used to simulate ground-truth responses in the model recovery experiments. For the sake of brevity, the validation performance described below is for the first instance. Drop-out was used during training, but disabled during inference (i.e., when testing the neural networks on held-out data or using them as representational models).

Each network was trained on one of three data sets: ImageNet [19], VGGFace [20], or synthetic faces we sampled and rendered from the Basel Face Model. For each data set, the mean and the standard deviation of the pixel intensity levels were estimated for the three color components of each image and then averaged across a sample of the data set's training images. These estimates were used to shift and scale the color channels of the input image to ensure expected zero mean and unit standard deviation inputs during training. The same intensity normalization was applied when the neural networks were used in stimulus optimization and data analysis.

Object recognition (ImageNet). We trained a VGG16 network on a 1000-way object classification task using the ImageNet data set. The images were preprocessed and augmented using datamodules.ImagenetDataModule, which is implemented in Lightning Bolts [30]. The training hyperparameters were as in the pretrained torchvision model reference script (`https://github.com/pytorch/vision/tree/main/references/classification`). We trained the network using stochastic gradient descent (SGD) with a weight decay of 0.0001, momentum of 0.9, and mini-batches of 1,024 images, minimizing the cross-entropy loss. The learning rate was initialized to 0.1 and decreased by a factor of 10 every 30 epochs. During both training and validation, each image was rescaled such that the shorter side was 146 pixels and then resized to 128 $\times$ 128 pixels using a random crop during training and a centered crop during validation. During training, the images were also augmented by random horizontal flip, applied with a probability of 0.2. The network was trained for 90 epochs and achieved 1.56 validation loss and 62.4% top-1 and 84.3% top-5 validation accuracy.

Face identification (VGGFace2). We trained a network of the same VGG-16 architecture to classify 8,631 face identities in the VGGFace2 dataset, minimizing the cross-entropy loss. Images were cropped using the data set's predefined bounding boxes and then rescaled and cropped again (random cropping during training and centered cropping during validation). Additionally, training images were augmented with greyscale transformation with a probability of 0.2. For each identity, we held 10 % of the images from the training data for validation. We used SGD with a weight decay of 0.0001, momentum of 0.9, and mini-batches of 1,024 images. The learning rate was initialized to 0.01 and decreased by a factor of 10 every ten epochs. The network was trained for 30 epochs and achieved 0.310 validation loss and 95.7 % top-1 and 98.2 % top-5 validation accuracy.

Face identification (BFM). A third VGG-16 network was trained to identify 8,631 synthetic face identities (matching the number of identities in the VGGFace2 natural face photo data set). Here the synthetic face images were generated with the Basel Face Model. For each synthetic identity, all of the face images shared the same randomly sampled shape and texture latents but had different expression latents, pose, lighting direction, and lighting intensity. Within-identity variability was further increased with 16 naturalistic augmentations [31], including different forms of noise addition, cutout, blurring, and grayscale transformation. 363 images of each identity were generated to roughly match the total number of training images in the VGGFace2 dataset [20]. The rendered images were preprocessed similarly to our VGGFace2 face identification model, with random cropping during training and centered cropping during validation. The training (minimizing the cross-entropy loss) was carried out by SGD with a weight decay of 0.0001, momentum of 0.9, and mini-batches of 512 images. The learning rate was initialized to 0.01 and reduced by a factor of 10 every ten epochs. The

model was trained for 30 epochs and reached a 0.00169 validation loss and 99.95 % top-1 and 99.99 % top-5 classification accuracy for the validation set.

Autoencoding (VGGFace2). We trained a variational autoencoder [VAE, 32, 33] on the VGGFace2 dataset. The autoencoder consisted of a VGG-16 as an encoder and an "upconvolutional" decoder. The penultimate layer of the VGG-16 was mapped by two fully connected layers to a predicted mean of a 500-dimensional Gaussian latent representation and to a variance parameter for each dimension, defining a diagonal covariance matrix. During training, the 500-dimensional latent vector was sampled according to the predicted mean and covariance and was then transformed back to an image by the decoder. The decoder network mapped the latent from its 500-dimensional space to a stack of 256 8×8 activation maps. These maps were then upsampled by four trainable upconvolutional blocks. Each block doubled the size of the maps and halved the number of channels. Each block included a 3×3 convolutional layer followed by a PixelShuffle upsampling operation, ReLU non-linearity, an additional 3×3 convolutional layer that maintained the activation dimensionality, and a second ReLU non-linearity. After the last block, the resulting 16 128×128 maps were mapped by a 3×3 convolutional layer and a sigmoid non-linearity to the original RGB image format. The VAE was trained to minimize the evidence lower bound (ELBO). The β hyper-parameter (the trade-off between reconstruction error and prior regularization) was set to 1.0 and the σ hyper-parameter (the scaling of the reconstruction error) was empirically fitted during training to minimize the ELBO [i.e., we used an "σ-VAE", 33]. We trained the network with ADAM [34] with $\beta_0 = 0.9$, $\beta_1 = 0.999$, $\epsilon = 10^{-4}$, a weight decay of 0.00001, and mini-batches of 256 images. The learning rate was initialized to 0.0005 and was reduced by a factor of 10 after 20 epochs. After training for 30 epochs, the network achieved 0.679 MSE for validation images. Visually, the reconstructed validation images had high fidelity. After training was completed, we discarded the decoder and used only the VGG-16 encoder as our representational model. For the ultimate layer, we used only the predicted mean of the latent representation without sampling from the predicted Gaussian in latent space.

Autoencoding (BFM). We repeated the VAE training procedure with the dataset of BFM synthetic faces described in the section Face identification (BFM) above. After training for 30 epochs, the network achieved 0.554 MSE for validation images. Here as well, the reconstructions had high visual fidelity.

Inverse rendering (BFM). Inspired by the Efficient Inverse Graphics (EIG) model by Yildirim et al. [27], we trained a VGG-16 to predict BFM face latents and extrinsic properties–pose and lighting. In the EIG model, different intermediate layers were trained to fit different representational targets, ranging from image segmentation to identity. Here, we trained a neural network to map each input image to a single output vector consisting of six components: 199 shape latents, 199 texture latents, 100 expression latents, four face pose parameters (expressed as a quaternion), three lighting color and three lighting direction parameters. This target representation resembles the EIG's f_5 representation, which was found by [27] to best match the macaque anterior face patch AM. We generated 3,300,000 unique synthetic faces using BFM by randomly sampling face latents, pose, lighting color, and direction. The loss function was defined as the sum of the normalized mean squared error (NMSE) for the six components. To maintain the veracity of the training targets, the only data augmentation during training was random cropping. We trained the network with ADAM [34] with $\beta_0 = 0.9$, $\beta_1 = 0.999$, $\epsilon = 10^{-8}$, no weight decay, and mini-batches of 512 images. The learning rate was initialized to 0.0001 and was reduced by a factor of 10 every 40 epochs. After training for 120 epochs, the model achieved 2.31 NMSE loss for validation images (where 0.0 is perfect performance and 6.0 is the error of the best constant predictor). Feeding the inferred latent representations back to the BFM renderer generated face images highly consistent with the input images.

A.2 Basel Face Model

The Basel Face Model (BFM) was conceived as a PCA-based probabilistic graphics model of faces [35] that combines 3D shape and mapped texture components to render nearly photorealistic images of human faces. The BFM characterizes each face by a shape, a texture, and (optionally) an expression, using separate latent vectors for each of these three determinants of face appearance. The shape of a face is parameterized by a latent vector $\boldsymbol{\alpha}$, defined such that $\mathbf{s}(\boldsymbol{\alpha}) = \bar{\mathbf{s}} + \sum_{i=1}^{r} \alpha_i \sqrt{\lambda_i} \mathbf{u}_i$, where $\mathbf{s}(\boldsymbol{\alpha})$ is a vector of vertex coordinates defining a 3D face mesh, $\bar{\mathbf{s}}$ is the vector of vertex coordinates of the average face mesh, λ_i is the i-th shape eigenvalue, and $\mathbf{u}_i$ is the i-th shape eigenvector. Texture components are handled analogously, enabling general linear mixing of 3D face models. The shape

and texture components define a normal distribution in latent space, which in the 2017 version was based on 3D scans of 200 people (not randomly sampled to be representative of any particular population). The normal distribution model enables us to sample faces at random by drawing the parameters $\alpha_i \sim \mathcal{N}(0, 1)$ or to evaluate the likelihood of a face with respect to that distribution. Note, however, that this distribution is not representative of the human population. In our optimization procedure, we adjust the latent vectors for both the shape and texture models. See [21] for a detailed mathematical description of the more recent BFM versions, where the PCA was generalized to a truncated Karhunen–Loève expansion over face deformations.

A.3 Stimulus optimization procedure

The synthesized stimuli were initialized as shape and texture BFM latent vectors randomly sampled from the BFM normal distribution model and then projected into a hyper-cube subtending $\pm$ 2 SDs. The face latents were then iteratively adjusted to maximize the optimization objective (Eq. 2).

Note that the stimulus optimization objective in Eq. 2 considers two trained model instances per model architecture and objective. This approach serves to safeguard against confusion of random variability across instances of a model trained from different random seeds for variability between different model architectures and objectives. For conceptual clarity, we defined one instance for each model as the "reference instance", denoting its predictions as $\hat{\mathbf{y}}^m$. This instance for each model was also used in data analysis. The other instance was used as the ground-truth model during the stimulus optimization, and its predictions are denoted by $\tilde{\mathbf{y}}^m$. The optimization objective in Eq. 2 considers only correlations *between* the reference and ground-truth instances. Note, however, that our stimulus-synthesis procedure can be improved by also considering the correlations between distinct models within each instance set. Furthermore, the minimal setup of two instances per model can be extended to a larger instance sample at the cost of greater GPU compute requirements.

We used the Adam optimizer with a learning rate set to 0.01. In this context, the learning rate controls the speed of face latent adjustment. In each optimization iteration, we evaluated the optimization objective ten times with jittered versions of the stimuli. We used a differentiable scale jitter between 99.5% and 100.5%, followed by a differentiable translation jitter of up to 10 pixels. After averaging the objective across these ten presentations, we took an optimization step ascending the gradient. Reparameterization kept the optimized face latents within the hyper-cube constraint: we parameterized each face latent dimension α_i by $\alpha_i = 2\tanh(\alpha_i')$, optimizing $\boldsymbol{\alpha}$ and using $\boldsymbol{\alpha}$, which lies in $(-2, 2)$, to generate faces. The learning rate was reduced by a factor of 2 after apparent convergence. Apparent convergence was defined as no significant improvement of the mean objective in the last 20 iterations compared to the mean objective in the preceding 20 iterations. Learning rate reduction was carried out twice. After the third apparent convergence (or the 1000-th iteration, whichever happened first), the optimization was terminated.

We used a compute server with eight GeForce RTX 2080 Ti GPUs. Two GPUs were devoted to PyTorch3D image rendering and the other six to model evaluation (we evaluated 144 images on 12 neural networks: 2 instances $\times$ 6 models). We used activation checkpointing to reduce GPU RAM requirements.

A.4 Human experiment screening criteria and participation compensation

Screening criteria. We took several steps to ensure that the participants recruited through prolific.co understood the arrangement task and made a sincere effort. A participant was excluded from the data analysis if they failed to meet any of the three criteria:

- Successful completion of a practice trial. In the practice trial, the participants arranged image pairs of various objects. We tested whether each participant's arrangement conformed to all of the following dissimilarity relations:
  1. (dog, strawberry) > (apple, orange)
  2. (car, apple) > (apple, orange)
  3. (apple, orange) > (apple, apple)
  4. (apple, orange)> (strawberry, strawberry).
- Low within-subject reliability. Since each participant was presented with every trial twice, we could estimate within-subject reliability by correlating the dissimilarity ratings in the two

repetitions, yielding one (Speamran's τ) correlation coefficient per trial. For each participant, we tested the significance of the Fisher-transformed correlation coefficients by a one-sample t-test against zero. Participants with $p \geq .05$ were disqualified for low reliability.

- Short participation duration. Participants who spent less than 10 minutes doing the experiment were excluded from analyses.

Participation compensation. We paid $816 in total to 136 participants, including 30 replication study participants and 16 disqualified participants who failed one the screen criteria. Three participants failed more than one screening criteria and were not paid. The hourly payment was $15.4, on average.

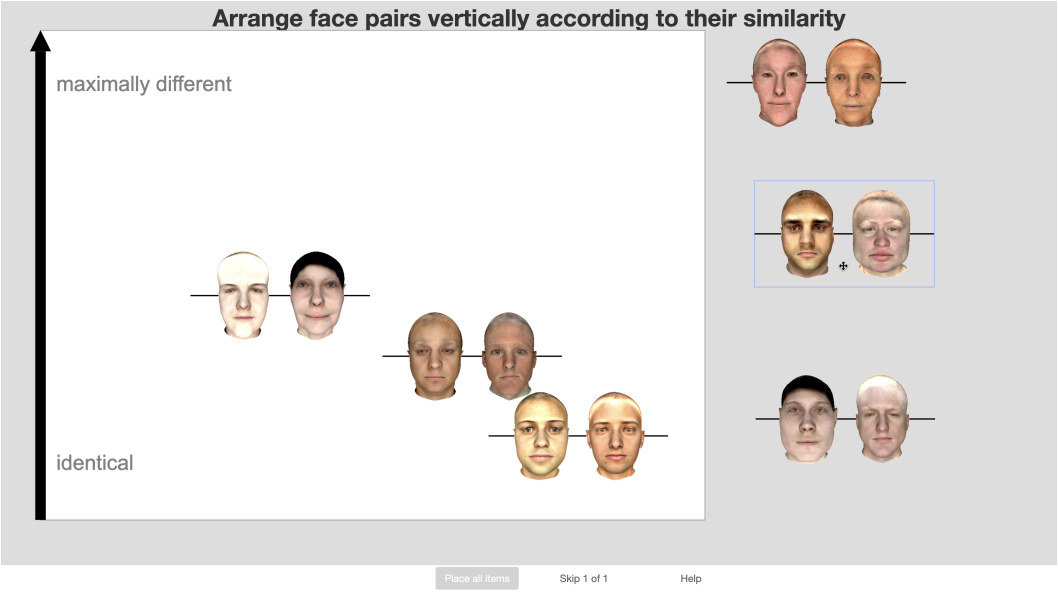

Figure S1: **An example experimental trial.** After completing an object-pair arrangement task for practice, participants were shown an annotated illustration of the face-pair arrangement task, with the following instruction alongside:

> In the following task, you will arrange face pairs vertically according to their similarity. In each trial, each pair of faces that are relatively more dissimilar from each other should be placed above each pair of faces that is relatively more similar to each other. Note that only the vertical positioning of faces will be taken into account. Horizontal space is provided so you can arrange the face pairs more easily.

For each arrangement screen, participants dragged each face pair from the grey area to the white arena and arranged the pairs vertically to indicate their perceptual similarity.

Note that the high prevalence of Caucasian faces reflects the particular sample underlying the Basel Face Model. Using alternative generators (or modified BFM latent distributions) can enable tuning the ethnic composition of the faces to better fit particular human populations.

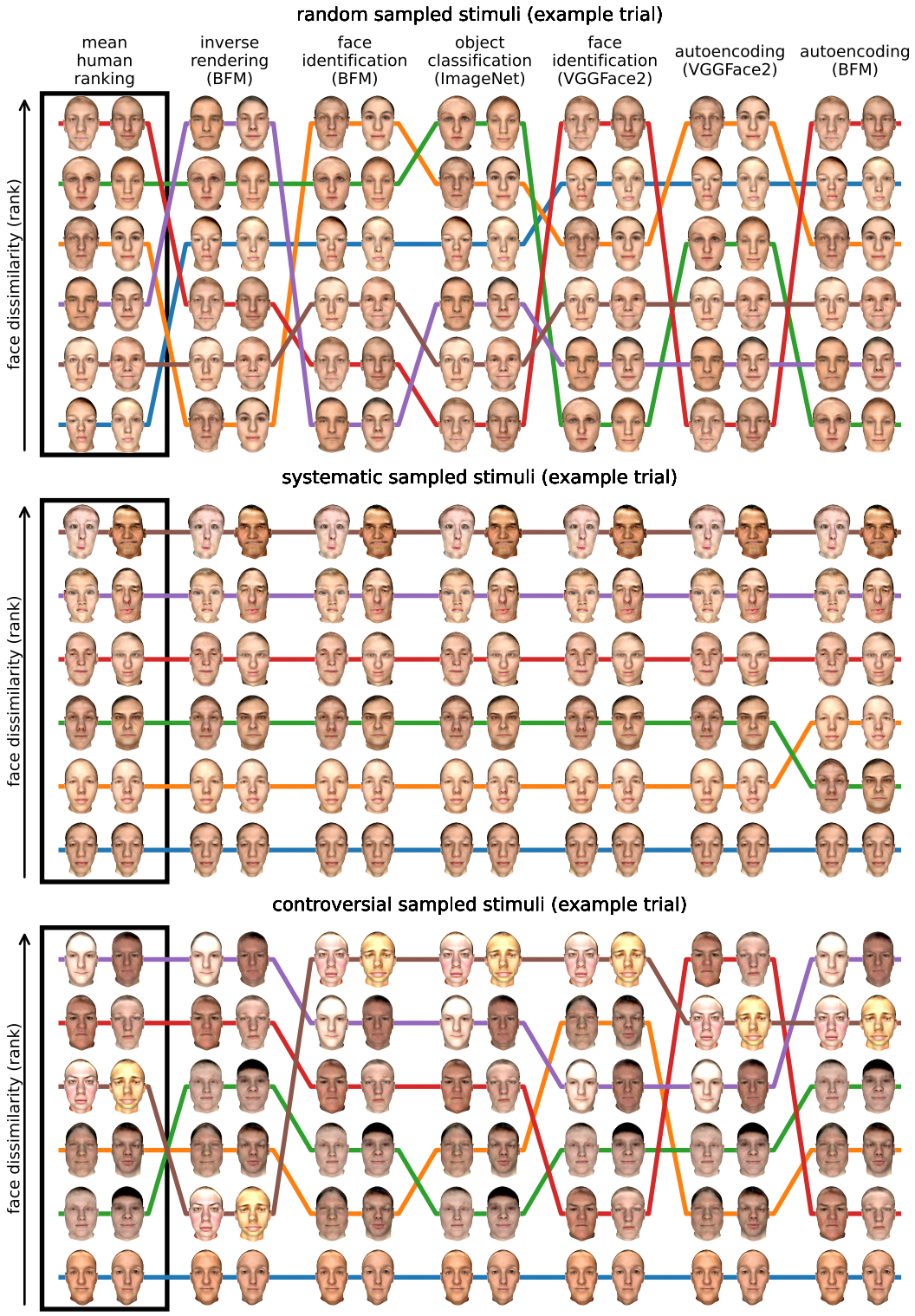

Figure S2: **Stimuli and dissimilarity rankings in one randomly sampled trial from each stimulus set: random, systematic and controversial stimuli.** Each colored line indicates the dissimilarity ranking of one face pair. The average human ranking is shown on the left (the ranked average across all participants' ranked dissimilarity ratings). The ranked predictions of each of the neural network models are shown on the right. For each model, we show here the predictions of its best-performing layer (see Fig. 2, top row), which is not necessarily the one that best predicts these particular trials.

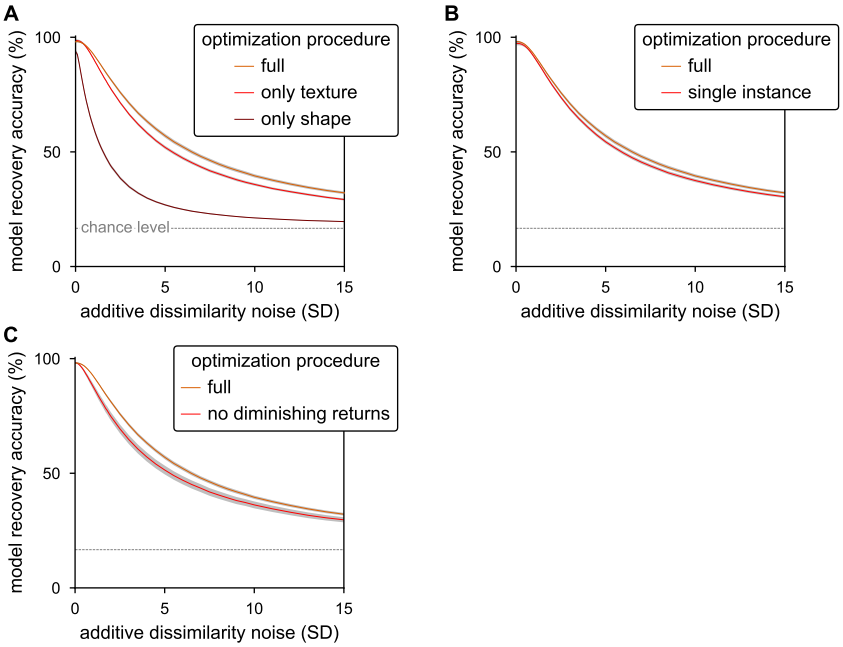

Figure S3: **Stimulus optimization ablation study.** We generated controversial stimulus sets either with the full optimization procedure described in the main text (orange-brown) or with various components turned off (red hues). We simulated model recovery accuracy for each stimulus set (see Fig. 1) and averaged the results across 10 stimulus sets of each optimization condition (each stimulus set was generated from a different random initialization of the face stimuli). The curves depict the mean model recovery accuracy as a function of simulated noise. The gray shaded regions depict ± 1 SE of this measure. **(A)** Optimizing only the texture component of the faces while keeping the randomly sampled shape component fixed leads to a slightly reduced model recovery accuracy compared to optimizing both components. Optimizing only the shape component leads to a dramatic reduction in model recovery accuracy. **(B)** Using only a single instance of each neural network model leads to a small reduction in model recovery accuracy. In this setting, the first term in Eq. 2 is constant, and the optimization focuses on features that are more idiosyncratic to the particular weight initializations of the neural network instances used (see [36]). **(C)** Setting the function $f(\cdot)$ in Eq. 2 to identity (i.e., $f(x) = x$ instead of $f(x) = -e^{-10x}$) reduces model recovery accuracy and increases the variability of this measure across repeated stimulus optimization runs. When $f(x)$ is set to identity, there are no diminishing returns for further separating a pair of representations that are already well-separated compared to other representation pairs.

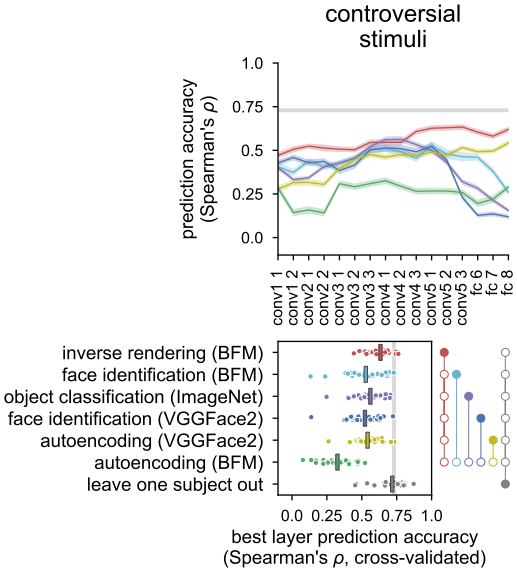

Figure S4: **A replication of the controversial stimuli experiment using a different controversial stimulus set.** Conventions are as in Figure 2. Since the controversial stimulus set was a product of random initialization followed by local optimization, we assessed the replicability of the results arising from using such stimulus sets by testing another group of 30 subjects with another controversial set, initialized with a different random seed. We found a result pattern consistent with the first controversial stimulus set we tested (Fig. 2, right column). Importantly, the neural network trained to invert the rendering of 2D face images performed significantly better than all other models in this replication.