The Perils and Pitfalls of Block Design for EEG Classification Experiments

Ren Li, Jared S. Johansen, Hamad Ahmed, Thomas V. Ilyevsky, Ronnie B. Wilbur, Hari M. Bharadwaj, and Jeffrey Mark Siskind, *Senior Member, IEEE*

Abstract—A recent paper [1] claims to classify brain processing evoked in subjects watching ImageNet stimuli as measured with EEG and to employ a representation derived from this processing to construct a novel object classifier. That paper, together with a series of subsequent papers [2], [3], [4], [5], [6], [7], [8], claims to achieve successful results on a wide variety of computer-vision tasks, including object classification, transfer learning, and generation of images depicting human perception and thought using brain-derived representations measured through EEG. Our novel experiments and analyses demonstrate that their results crucially depend on the block design that they employ, where all stimuli of a given class are presented together, and fail with a rapid-event design, where stimuli of different classes are randomly intermixed. The block design leads to classification of arbitrary brain states based on block-level temporal correlations that are known to exist in all EEG data, rather than stimulus-related activity. Because every trial in their test sets comes from the same block as many trials in the corresponding training sets, their block design thus leads to classifying arbitrary temporal artifacts of the data instead of stimulus-related activity. This invalidates all subsequent analyses performed on this data in multiple published papers and calls into question all of the reported results. We further show that a novel object classifier constructed with a random codebook performs as well as or better than a novel object classifier constructed with the representation extracted from EEG data, suggesting that the performance of their classifier constructed with a representation extracted from EEG data does not benefit from the brain-derived representation. Together, our results illustrate the far-reaching implications of the temporal autocorrelations that exist in all neuroimaging data for classification experiments. Further, our results calibrate the underlying difficulty of the tasks involved and caution against overly optimistic, but incorrect, claims to the contrary.

Index Terms—Object classification, EEG, neuroimaging

1 Introduction

It is well known in the neuroimaging community that both fMRI and EEG time series exhibit temporal autocorrelations both in the short and long range regardless of experimental stimuli [9], [10]. Accordingly, to avoid confounding block-level effects with experimental effects, neuroscience studies employ designs that distribute each experimental condition across multiple blocks, or use temporally jittered stimuli to break the correlation structure, and/or use rapid-event designs where stimuli are randomized at a single-trial level [11], [12]. However, despite the explosion of studies using machine learning techniques applied to neuroimaging data [13], [14], to our knowledge, the effects of EEG/fMRI temporal correlations on classification problems have not been examined by the machine-learning community. Here, we illustrate the far-reaching implications of such temporal correlations in EEG data and the importance of adherence to rigorous experiment design considerations by comprehensively analyzing the seemingly impressive claims made by a recent paper [1] and through a series of additional experiments carefully designed to elucidate the issues.

A recent paper [1] claims to (learn to) classify EEG data recorded from human subjects observing images from ImageNet [15] and use the learned classifier to train a pure computer-vision model. In that paper, images from ImageNet are presented as stimuli to human subjects undergoing EEG and a long short-term memory (LSTM [16]), combined with a fully connected layer and a ReLU layer, is trained to predict the class of the stimulus from the recorded EEG signal. The output of the ReLU layer is taken to reflect human neural encoding of the percept. The output of existing object classifiers is then regressed to this purported human neural encoding of the percept in an attempt to have computer-vision systems produce the same encoding of the percept.

That paper makes three specific claims [1, Section 1 p. 6810]:

1. *We propose a deep learning approach to classify EEG data evoked by visual object stimuli outperforming state-of-the-art methods both in the number of tackled object classes and in classification accuracy.*
2. *We propose the first computer vision approach driven by brain signals, i.e., the first automated classification approach employing visual descriptors extracted directly from human neural processes involved in visual scene analysis.*

- *Ren Li, Jared S. Johansen, Hamad Ahmed, Thomas V. Ilyevsky, and Jeffrey Mark Siskind are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907. E-mail: tomo.blade.lee@hotmail.com, {jjohanse, ahmed90, tilyevsk, qobi}@purdue.edu.*
- *Ronnie B. Wilbur is with the Department of Speech, Language, and Hearing Sciences and the Department of Linguistics, Purdue University, West Lafayette, IN 47907. E-mail: wilbur@purdue.edu.*
- *Hari M. Bharadwaj is with the Weldon School of Biomedical Engineering and the Department of Speech, Language, and Hearing Sciences, Purdue University, West Lafayette, IN 47907. E-mail: hbharadwaj@purdue.edu.*

Manuscript received 19 Dec. 2018; revised 23 Nov. 2019; accepted 7 Feb. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Jeffrey Mark Siskind.)
Recommended for acceptance by J. DiCarlo.
Digital Object Identifier no. 10.1109/TPAMI.2020.2973153

3. *We will publicly release the largest EEG dataset for visual object analysis, with related source code and trained models.*

In particular, regarding claim 1, that paper further claims:

i. Their method can classify a far larger number (40) of distinct object classes than prior work (at most 12 [17], typically 2) on classifying EEG signals.
ii. Their method achieves far higher accuracy (82.9 percent) than prior work [17] (29 percent) on classifying EEG signals.

That paper further couches its purported results in superlatives:

> *In this paper, we want to take a great leap forward with respect to classic BCI approaches, i.e., we aim at exploring a new and direct form of human involvement (a new vision of the "human-based computation" strategy) for automated visual classification. The underlying idea is to learn a brain signal discriminative manifold of visual categories by classifying EEG signals—reading the mind–and then to project images into such manifold to allow machines to perform automatic visual categorization—transfer human visual capabilities to machines. The impact of decoding object category-related EEG signals for inclusion into computer vision methods is tremendous. First, identifying EEG-based discriminative features for visual categorization might provide meaningful insight about the human visual perception systems. As a consequence, it will greatly advance performance of BCI-based applications as well as enable a new form of brain-based image labeling. Second, effectively projecting images into a new biologically based manifold will change radically the way object classifiers are developed (mainly in terms of feature extraction). [1, Section 1 pp. 6809–6810].*

Here, we report a number of experiments and analyses that call these results and claims into question. Specifically, we find that the classifier employed makes extensive, if not sole, use of long-term static brain activity that persists much longer than the duration of individual stimuli. Since the paper employs a block design, where all stimuli of a given class are presented to a subject in succession, the classifiers employed tend to classify the brain activity during that block, which appears to be largely uncorrelated with stimulus class. This is exacerbated by the reliance of the classifier on DC and very-low frequency (VLF) components in the EEG signal that reflect arbitrary long-term static mental states during a block rather than dynamic brain activity. Since each trial in the test sets employed comes from the same block as many trials in the corresponding training sets, the reported high classification accuracy results from classifying arbitrary temporal artifacts of the data instead of stimulus-related activity. When the experiment is repeated with a rapid-event design, where stimuli of different classes are randomly intermixed, classification accuracy drops to chance. As a result, this renders suspect all of the results and claims advanced in multiple published papers [1], [2], [3], [4], [5], [6], [7], [8]. Our experiments suggest that the underlying tasks are far more difficult than they appear on the surface and are far beyond the current state of the art. This suggests caution in light of widely published [1], [2], [3], [4], [5], [6], [7], [8] sensational claims that are overly optimistic but incorrect. Finally, in Section 6, we scrutinize 122 recent papers that classify EEG data and find that a significant fraction are problematic in ways described here.

2 Overview

In Section 3, we report a comprehensive set of experiments and analyses to fully understand the results and claims reported by Spampinato *et al.* [1] (henceforth OP_1, "original paper"). We first summarize our findings:

a. In Section 3.3, we reanalyze the EEG data collected by OP_1 using a number of different classifiers in addition to the one based on an LSTM that was employed by OP_1. We show that one can obtain good, if not better, results with other classifiers, particularly ones that are sensitive to temporal alignment, unlike LSTMs. When we further reanalyze the EEG data collected by OP_1 with shorter temporal windows (as short as a single temporal sample), with random temporal offset, and with a reduced set of channels, we obtain even better results with these different classifiers. This suggests that the data collected by OP_1 lacks temporal and detailed spatial information reflective of perceptual processes that would benefit classification.
b. In Section 3.4, we replicate the data collection effort of OP_1 using the same stimuli, presentation order, and timing, recording 96 channels with finer quantization (24 *versus* 16 bits) and higher temporal sampling rate (4096 Hz *versus* 1 kHz).[1] We do this both with the original block design employed by OP_1, where all stimuli of a given class are presented together, and with a rapid-event design, where stimuli of different classes are randomly intermixed. We also collect data with both the block and rapid-event designs, both for the original still-image stimuli depicting objects from ImageNet and short video clips depicting activity classes from Hollywood 2 [18].
c. In Section 3.5, we replicate all of the analyses of Section 3.3 on our new data. For data collected with the block design, we obtain moderately good classification accuracy on both image and video stimuli with one classifier, long windows, and a large set of channels. However, we obtain poor classification accuracy with all of the other classifiers, shorter windows, and a small set of channels. We further find that all classifiers yield chance performance on data collected with a rapid-event design.
d. OP_1 state that their data analysis included bandpass and notch filtering. Thus the analyses in Section 3.5 employed such filtering, which removes the DC and VLF components. Since the authors of OP_1 indicated to us in email (Section 4.1) that they did not perform bandpass filtering, in Section 3.6, we repeat the analysis of our data without bandpass filtering as well. Retaining the DC and VLF

1. For compatibility with the analyses of OP_1, we downsample our data to 1024 Hz. Nonetheless, we release our raw data with the 4096 Hz sample rate for future use.

component allows us to replicate the results obtained on the data released by OP_1 with our data collected with a block design. However, we still obtain chance for our data collected with a rapid-event design.

e. The block design employed by OP_1, together with their splits, has the property that every trial in each test set comes from a block that contains many trials in the corresponding training set. In Section 3.7, we conduct four new analyses. In the first new analysis, we repeat the analysis on the data released by OP_1 using new splits where the trials in each test set come from blocks that do not contain trials in the corresponding training set. Classification accuracy drops to chance. In the second new analysis, we attempt within-subject cross-block classification on our data collected with a block design. Since we recorded three separate runs with a block design for both image and video stimuli, from the same subject, two with the same stimulus presentation order and one with a different stimulus presentation order, we are able to conduct cross-block analyses where the trials in the test set come from different blocks than those in the corresponding training set. We first attempt cross-block classification between block runs with the same stimulus presentation order. Classification accuracy drops precipitously when the data is not preprocessed with a bandpass filter. Further, when the data has been preprocessed with a bandpass filter, classification accuracy drops to chance. Finally, when attempting cross-block classification between block runs with different stimulus presentation orders, classification accuracy drops to chance even when the data is not preprocessed with a bandpass filter. In the third new analysis, we repeat the analysis on our new data collected with a rapid-event design, where the labels are replaced with arbitrary labels that are correlated with the block instead of the stimulus. Classification accuracy rises from chance to levels far above chance, reaching those obtained on the data collected by OP_1. In the fourth new analysis, we rerun the code released by OP_1 on the data released by OP_1 after first applying various highpass filters to the data. Classification accuracy drops from roughly 93 percent to roughly 32 percent. Collectively these demonstrate that the high classification accuracies reported by OP_1 result from classifying the long-term brain activity associated with a block, even when that block contains stimuli of different classes, not the brain activity associated with perception of the class of the stimuli. They further demonstrate that this is exacerbated by the presence of DC and VLF components of the signal that remain due to lack of bandpass filtering. This refutes claims 1 and 3.

f. In Sections 3.8 and 3.9, we replicate the regression and transfer-learning analyses performed by Spampinato *et al.* [1, Sections 3.3, 4.2, and 4.3] but with a twist. We replace the EEG encodings with a random codebook and achieve equivalent, if not better, results. This demonstrates that the regression and transfer-learning analyses performed by OP_1 are not benefiting from a brain-inspired or brain-derived representation in any way, refuting claim 2.

3 Experiments

Our findings in Sections 5 and 7 are supported by the following experiments and analyses performed.

3.1 The OP_1 Data Collection

OP_1 adopted the following experimental protocol. They selected 40 object classes from ImageNet [1, footnote 1] along with 50 images for each class. These were presented as stimuli to 6 human subjects undergoing EEG. A block design was employed. Each subject saw 40 blocks, each containing 50 image stimuli. Each image was presented exactly once. All 50 stimuli in a block were images of the same class. All subjects saw exactly the same 2,000 images. We do not know whether different subjects saw the classes, or the images in a class, in different orders. The image presentation order for one subject was provided to us by the authors. Each image was presented for 0.5 s. Blocks were separated by 10 s of blanking. Approximately $40 \times (50 \times 0.5 \text{ s} + 10 \text{ s}) = 1400 \text{ s}$ of EEG data were collected from 128 channels at 1 kHz with 16 bit resolution.

3.2 The OP_1 Data Analysis

OP_1 report that the EEG data was preprocessed by application of a second-order bandpass Butterworth filter (low cut-off frequency 14 Hz, high cut-off frequency 71 Hz) and a notch filter (49–51 Hz). The pass band was selected to include *the Beta (15–31 Hz) and Gamma (32–70 Hz) bands, as they convey information about the cognitive processes involved in the visual perception* [1, Section 3.1 p. 6812]. The data for all 6 subjects was pooled, segmented into trials of approximately 0.5 s duration, and divided into six training/validation/test-set splits. Each portion of each split contained data from all 6 subjects and all classes for all subjects. The data was z-scored prior to training and classification. An LSTM, combined with a fully connected layer and a ReLU layer, was applied to a 440 ms window of each trial starting 40 ms from stimulus onset. A variety of different architectural parameters were evaluated, the best of which achieved 85.4 percent validation accuracy and 82.9 percent test accuracy. OP_1 claim that this is significantly higher classification accuracy for a significantly larger number of classes than all prior reported classification experiments on EEG data [17], [19], [20], [21], [22], [23], [24], [25], [26].

3.3 Reanalysis of the OP_1 Data

We asked whether the significant improvement in classification ability was due to the classifier architecture employed by OP_1 or whether it was due to some aspect of their experimental protocol and data collection procedure. OP_1 have publicly released their code[2] and data.[3] This allowed us to

2. http://perceive.dieei.unict.it/files/cvpr_2017_eeg_encoder.py
3. http://perceive.dieei.unict.it/index-dataset.php?name=EEG_Data

TABLE 1
Classification Accuracy Averaged Across Validation Sets, Test Sets, and All Six Splits Used by OP_1 on Their Released Data With Their Software (an LSTM Combined With A Fully Connected Layer and a ReLU Layer) and Four New Classifiers: A Nearest Neighbor Classifier (k-NN), an SVM, an MLP, and a 1D CNN

LSTM	k-NN	SVM	MLP	1D CNN
94.7%*	42.2%*	94.4%*	45.8%*	96.7%*

Here, and in all tables, starred values indicate above chance ($p < 0.005$) by a binomial cmf.

verify their published results and to reanalyze their data with different classifiers to investigate this question. The released code yields (slightly better than) the published accuracy on the released data.

OP_1 have released their data in both Python and Matlab formats. Both formats are subsequent to segmentation. All results reported here were produced with the Python format data which was z-scored before processing. See Section 4 for details.

We reanalyzed the OP_1 data with four different classifiers (Table 1): a k-nearest neighbor classifier (k-NN), a support vector machine (SVM [27]), a multilayer perceptron (MLP), and a 1D convolutional neural network (CNN).[4] The k-nearest-neighbor classifier used $k = 7$ with a Euclidean distance on the $128 \times 440 = 56320$ element vector associated with each trial. The SVM employed a linear kernel applied to data that was temporally downsampled to 500 Hz, i.e., $128 \times 220 = 28160$ element vectors. The MLP employed two fully connected layers with a sigmoid activation function after the first fully connected layer, and no dropout, trained with a cross-entropy loss, applied to $128 \times 440 = 56320$ element vectors, with 128 hidden units. The 1D CNN (Fig. 1) processed each of the 128 channels independently with eight 1D CNNs of length 32 and stride 1. The 128 applications of each of the eight 1D CNNs shared the same parameters. The output of each was processed by an ELU, followed by dropout with probability of 0.5. This yielded a temporal feature stream of length $440 - 32 + 1 = 409$ with $128 \times 8 = 1024$ features per time point. This was then processed by a fully connected layer mapping each time point to a 40 element vector. The parameters were shared across all time points. This was then processed by average pooling along the time axis, independently for each of the 40 channels, with a kernel of length 128 and a stride of 64. This produced a feature map with 40 features for 5 time points. Dropout with probability 0.5 was then applied, followed by a fully connected layer with 40 outputs. Training was performed with a cross-entropy loss. For the LSTM, temporal EEG samples for a trial were provided one-by-one as input to the classifier. For the 1D CNN, a matrix whose rows were channels and whose columns were temporal EEG samples for a trial was presented as input to the classifier. For the other classifiers, all temporal EEG samples for a trial were concatenated and presented as a single input vector.

4. All code and raw data needed to replicate the results in this paper are available at http://dx.doi.org/10.21227/x2gf-5324

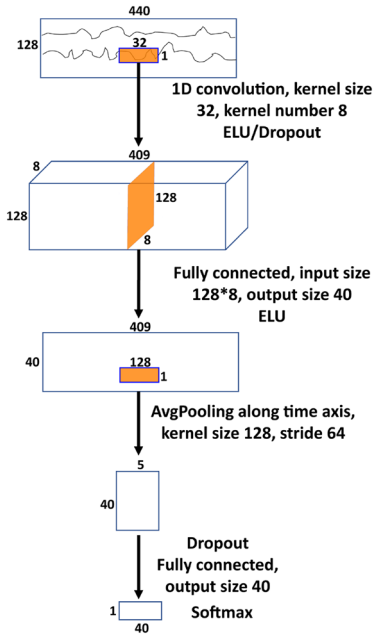

Fig. 1. Our 1D CNN used to process EEG data.

The results in Table 1 suggest that there is nothing specific about the classifier architecture employed by OP_1 that yields high results. The same results can be obtained not only with an LSTM-based classifier or a 1D CNN that attempts to model the temporal nature of the signal, but also with an SVM that has no particular temporal structure. Moreover, while other methods such as k-NN and MLP that also lack temporal structure do not yield as high accuracy, they nonetheless yield accuracy far higher than chance and far higher than any of the results reported in the literature cited by OP_1: [17], [19], [20], [21], [22], [23], [24], [25], [26].

Given that high accuracy was achieved with classifiers that should be sensitive to temporal translation of the signal, we asked whether the classification accuracy depended on this. To this end, we trained and tested all 5 classifiers, varying the length of the trial window between 200 ms, 100 ms, 50 ms, and 1 ms (Table 2).[5] In all cases, the trial window was started at a random offset from the stimulus onset, on a trial-by-trial basis. Note that high accuracy can even be obtained with a single temporal sample randomly selected within the stimulus interval. This suggests that no temporal brain processing is reflected in the classification accuracy.

An earlier report [2] conducted a similar data collection effort to that of OP_1 with 32 channels instead of 128. That effort yielded considerably lower classification accuracy (about 40 percent) on the same classes, stimuli, experimental protocol, and classification architecture. Given that the classifiers analyzed here appear not to rely on the temporal nature of brain processing, we asked

5. Several relevant architectural parameters of some of the classifiers vary from those presented in Section 3.3 for different window lengths. Due to the nature of its design, the 1D CNN model was never applied to windows shorter than 200 ms. Further, when running the SVM on 440 ms windows, the data was downsampled to 500 Hz, as per Section 3.3, but on all other window sizes, the data was processed at 1 kHz.

TABLE 2
Classification Accuracy for Varying Trial Window Lengths With Random Temporal Offset From the Stimulus Onset, Averaged Across Validation Sets, Test Sets, and All Six Splits Used by OP_1 on Their Data With All Five Classifiers

window	LSTM	k-NN	SVM	MLP	1D CNN
200 ms	93.9%*	38.8%*	94.1%*	61.0%*	97.4%*
100 ms	94.3%*	37.9%*	94.0%*	76.4%*	n/a
50 ms	94.7%*	38.4%*	94.2%*	85.2%*	n/a
1 ms	93.4%*	42.3%*	90.9%*	92.4%*	n/a

how much they rely on the number of channels. To this end, we performed feature, i.e., channel selection on the dataset to train and test with various subsets of channels of different sizes. The Fisher score [28] of a channel v for a classification task with C classes where each class c has n_c examples was computed as

$$\frac{\sum_{c=1}^{C} n_c(\mu_{c,v} - \mu_v)^2}{\sum_{c=1}^{C} n_c \sigma_{c,v}^2}, \tag{1}$$

where $\mu_{c,v}$ and $\sigma_{c,v}$ were the per-class per-channel means and variances and μ_v was the per-channel mean. We selected the m channels with highest Fisher score on the training set, for varying m, and repeated the training and testing on this subset of channels for varying window lengths (Table 3).[6] We observe that the full 128 channels are not necessary to achieve high accuracy. While the accuracy degrades somewhat when using fewer than 32 channels, one can obtain far greater accuracy than chance and far greater accuracy than all prior reported classification experiments cited by OP_1: [17], [19], [20], [21], [22], [23], [24], [25], [26] on EEG data with as few as 8 channels. Moreover, one can obtain far greater accuracy than Spampinato *et al.* [2] with the same number (32) of channels and the same accuracy with far fewer (8) channels. While the spatial layout of channel selection might not coincide with the electrode placement of a cap with fewer electrodes, we next discuss why we consider it important that one can accurately classify the OP_1 data with such extreme spatial and temporal downsampling.

3.4 New Data Collection

The above analyses suggest that the accuracy achieved by OP_1 was not due to the analysis architecture but rather due to either the experimental protocol (block design, stimuli, and stimulus timing and presentation order) or the data collection effort (their laboratory apparatus—caps and acquisition hardware—used). We asked whether the accuracy was due to the former or the latter. To this end, we repeated the data collection effort. We collected data from six subjects. For each, we collected four kinds of data. The first two used the same 40 object classes and 2,000 image stimuli as OP_1. The second two used the 12 activity classes and a subset of the video clips from Hollywood 2 as described in Siskind [29]. The subset of clips

6. Several relevant architectural parameters of some of the classifiers vary from those presented in Section 3.3 for different numbers of channels.

TABLE 3
Classification Accuracy for Varying Numbers of Channels, Averaged Across Validation Sets, Test Sets, and All Six Splits Used by OP_1 on Their Data With All Five Classifiers and Varying Trial Window Lengths With Random Temporal Offset From the Stimulus Onset

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	94.3%*	44.7%*	95.6%*	60.6%*	97.5%*
200 ms	96	96.0%*	41.4%*	95.2%*	75.4%*	97.9%*
100 ms	96	95.2%*	41.3%*	95.0%*	85.1%*	n/a
50 ms	96	97.4%*	41.3%*	94.9%*	89.5%*	n/a
1 ms	96	93.9%*	48.0%*	91.0%*	92.6%*	n/a
440 ms	64	96.3%*	54.6%*	95.9%*	74.3%*	96.6%*
200 ms	64	97.0%*	51.8%*	95.2%*	85.7%*	97.1%*
100 ms	64	96.8%*	52.1%*	95.0%*	90.1%*	n/a
50 ms	64	97.3%*	52.3%*	95.1%*	92.6%*	n/a
1 ms	64	95.3%*	61.5%*	90.5%*	92.5%*	n/a
440 ms	32	83.5%*	57.1%*	84.4%*	83.7%*	89.4%*
200 ms	32	91.7%*	53.9%*	81.3%*	87.1%*	89.3%*
100 ms	32	95.2%*	54.1%*	81.2%*	88.7%*	n/a
50 ms	32	95.8%*	54.0%*	80.8%*	90.9%*	n/a
1 ms	32	79.2%*	61.8%*	65.9%*	81.4%*	n/a
440 ms	24	66.9%*	55.0%*	70.5%*	81.2%*	81.8%*
200 ms	24	66.4%*	51.3%*	66.3%*	82.2%*	81.2%*
100 ms	24	92.0%*	51.8%*	65.7%*	83.6%*	n/a
50 ms	24	91.6%*	51.9%*	64.6%*	84.7%*	n/a
1 ms	24	80.7%*	56.8%*	48.9%*	71.0%*	n/a
440 ms	16	43.6%*	56.2%*	49.5%*	77.5%*	66.6%*
200 ms	16	72.2%*	54.2%*	44.5%*	77.4%*	67.1%*
100 ms	16	82.9%*	53.1%*	43.1%*	78.0%*	n/a
50 ms	16	85.1%*	52.8%*	42.2%*	78.8%*	n/a
1 ms	16	65.5%*	53.0%*	34.1%*	58.5%*	n/a
440 ms	8	31.5%*	47.5%*	17.2%*	58.1%*	39.9%*
200 ms	8	47.8%*	43.8%*	13.7%*	55.5%*	38.1%*
100 ms	8	66.9%*	44.2%*	12.7%*	60.2%*	n/a
50 ms	8	68.4%*	43.2%*	13.3%*	59.6%*	n/a
1 ms	8	42.5%*	41.8%*	14.0%*	36.6%*	n/a

was selected to be counterbalanced, with 32 clips per class, temporally cropped to a uniform 4 s duration centered around the activity class depicted, and transcoded to a uniform spatial and temporal resolution. We repeat all of our experiments and analyses on both image and video stimuli to investigate whether the issues that arise are particular to the task of classifying object perception (nouns) or whether they also arise in the task of classifying activity perception (verbs).

Data was collected with two different paradigms for each set of stimuli. One paradigm used a block design, where all stimuli of a given class were shown together in a single block. The other paradigm used a rapid-event design, where the stimuli were presented in randomized order.

For subject 1, we collected the block data once, thus collecting four recordings: one image block, one image rapid event, one video block, and one video rapid event. For subjects 2–5, we collected the block data twice, both with the same stimulus presentation order, thus collecting six recordings per subject: two image block, two video block, one image rapid event, and one video rapid event. For subject 6, we collected the block data three times, the first two with the same stimulus presentation order and the third with a different order. This alternate order varied both the order in which the classes were presented as blocks and the order in which the stimuli within a class were presented within a block. Thus for subject 6, we collected eight recordings: three image block, three video block, one image rapid event, and one video rapid event. The data for subject 1 was collected in two sessions (one capping for each), one for

image stimuli and one for video stimuli. The data for each remaining subject was collected in a single session with a single capping. Since all analyses on our data are within subject and just on images or just on video, no alignment was necessary.

The block design for the image stimuli employed the same design as OP_1: 40 blocks, each consisting of 50 stimuli, each presented for 0.5 s with 10 s of blanking after each block. For all but the third block run for subject 6, the presentation order of the classes and stimuli within each class were the same as in the data collected by OP_1.

The rapid-event design for the image stimuli also employed 40 blocks, each consisting of 50 stimuli, each presented for 0.5 s with 10 s of blanking after each block, just that each block contained a random selection of images from different classes. In the latter, different blocks could contain different numbers of images for different classes, subject to the constraint that, over the entire experiment, each of the 2,000 images was shown exactly once.

The block design for the video stimuli began with 8 s of fixation blanking, followed by 12 blocks, during each of which 32 clips were presented in succession, each lasting 4 s, with 10 s of fixation blanking after each block. Approximately $12 \times (32 \times 4 \text{ s} + 10 \text{ s}) = 1656 \text{ s}$ of EEG data were collected. For the block design, all stimuli within the block were of the same class. For all but the third block run for subject 6, the presentation order of the classes and stimuli within each class were the same.

The rapid-event design for the video stimuli also employed 12 blocks, each consisting of 32 stimuli, each presented for 4 s with 10 s of blanking after each block, just that each block contained a random selection of clips from different classes. In the latter, different blocks could contain different numbers of clips for different classes, subject to the constraint that, over the entire experiment, each of the 384 clips was shown exactly once.

Unlike the data collection effort of OP_1, which divided each recording into four 350 s sessions, each of our 36 recordings was collected in a single session. EEG data was recorded from 96 channels at 4,096 Hz with 24 bit resolution using a BioSemi ActiveTwo recorder[7] and a BioSemi gel electrode cap. Two additional channels were used to record the signal from the earlobes for rereferencing. A trigger was recorded in the EEG data to indicate stimulus onset. We downsampled the data to 1.024 kHz, rereferenced the data to the earlobes, and employed the same preprocessing as reported by OP_1: a bandpass filter (low cut-off frequency 14 Hz, high cut-off frequency 71 Hz), a notch filter (49–51 Hz), and z-scoring.[8]

3.5 Analysis of Our New Data

We applied the analysis from Table 3 to our new data collected with the block design for the image (Table 4 left) and video (Table 4 right) stimuli.[9] This subsumes all analyses performed on the OP_1 data. Note that we are not able to replicate the results of OP_1. While the 1D CNN achieves moderately good performance on both image and video stimuli, the other classifiers perform poorly. Moreover, for shorter analysis windows, random offsets, and reduced numbers of channels, the other classifiers perform largely at chance. We analyze the source of this difference below.

We then applied all of the classifiers from Table 1 to the data collected with a rapid-event design for the image (Table 5 left) and video (Table 5 right) stimuli. Note that all classifiers yield chance performance.

3.6 Spectral Analysis

We asked why it is possible to achieve high accuracy with short analysis windows on the OP_1 data but not with our data. The authors of OP_1 indicated to us in email that their report of preprocessing was a misprint and that they performed notch filtering (during acquisition) and z-scoring but not bandpass filtering. See Section 4.1 for details. Since their released code performs z-scoring, this indicates that their released data reflects notch filtering but neither bandpass filtering nor z-scoring. We thus reanalyzed our data with a notch filter and z-scoring but no bandpass filter (Tables 6 and 7). Note that we now obtain better results for the data collected with the block design, similar to that obtained with the data released by OP_1, but still obtain chance for data collected with the rapid-event design.

3.7 Block *Versus* Rapid-Event Design

We asked why we (and OP_1) are able to obtain high classification accuracy with a block design but not a rapid-event design. To this end, we performed four reanalyses.

First, we repeated the analysis from Tables 1, 2, and 3, where instead of using the training/test set splits provided by OP_1, we conducted a leave-one-subject-out round-robin cross validation, training on all data from five of the subjects and testing on all data from the sixth, rotating among all six subjects as test (Table 8). Note that classification accuracy is now at chance.

Second, we performed cross-block analyses on our new data, whereby we trained models on one block run from a

7. The ActiveTwo recorder employs 64× oversampling and a sigma-delta A/D converter, yielding less quantization noise than 24 bit uniform sampling.

8. OP_1 presumably applied a notch filter to remove 50 Hz line noise. Being in the US, we should nominally remove 60 Hz line noise instead of 50 Hz. However, after rereferencing, our data contains no line noise so notch filtering is unnecessary. We employ a 50 Hz notch filter just to replicate OP_1.

9. We present here and elsewhere in this paper the results for subject 6, often just for a subset of the block runs. The Appendix in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2973153, contains the results for all other block runs for subject 6 and for all block runs for all other subjects. All subjects and all block runs for all subjects exhibit the same broad pattern of results. Unlike the analyses of OP_1 which pool data from all six subjects into both the training and test sets, we do not pool data. Further, unlike the lone cross-subject reanalysis of the OP_1 data in Section 3.7, we do perform cross-subject train and test on our data. All analyses on our data train and test on data from the same subject. For purposes of calculating window lengths for our new data, we treat 1,024 Hz as 1 kHz. Several relevant architectural parameters of some of the classifiers vary from those presented in Section 3.3 when applied to video data, particularly with different window lengths and/or numbers of channels. Due to the nature of its design, the 1D CNN model was never applied to video data with windows shorter than 1,000 ms. Further, when running the SVM on video data, the data was downsampled to 500 Hz irrespective of window length.

TABLE 4
Application of the Analysis From Table 3 to the First Block Run of Subject 6 on (left) Image and (right) Video Stimuli, Where the Data has been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	16.5%*	2.1%	3.1%	3.3%	37.8%*
200 ms	96	13.7%*	3.1%	2.9%	2.8%	33.9%*
100 ms	96	13.7%*	3.1%	3.0%	3.4%	n/a
50 ms	96	9.2%*	4.4%*	3.0%	3.7%*	n/a
1 ms	96	5.5%*	4.7%*	3.0%	3.8%*	n/a
440 ms	64	11.9%*	2.7%	2.9%	3.3%	27.3%*
200 ms	64	9.4%*	2.9%	3.2%	2.8%	26.6%*
100 ms	64	9.2%*	2.9%	2.5%	2.5%	n/a
50 ms	64	8.7%*	3.4%	2.5%	2.6%	n/a
1 ms	64	4.7%*	5.1%*	2.7%	4.0%*	n/a
440 ms	32	8.6%*	2.7%	3.1%	3.7%*	21.7%*
200 ms	32	7.7%*	3.8%*	3.0%	2.8%	19.4%*
100 ms	32	7.3%*	3.4%	2.9%	3.1%	n/a
50 ms	32	6.1%*	3.1%	3.1%	2.5%	n/a
1 ms	32	6.1%*	4.0%*	2.6%	4.2%*	n/a
440 ms	24	9.9%*	3.0%	4.1%*	4.2%*	19.5%*
200 ms	24	5.6%*	3.4%	3.0%	3.4%	15.1%*
100 ms	24	4.5%*	2.9%	3.0%	2.7%	n/a
50 ms	24	4.9%*	3.1%	2.8%	2.5%	n/a
1 ms	24	4.6%*	3.6%*	2.0%	3.4%	n/a
440 ms	16	7.9%*	3.1%	4.0%*	5.5%*	17.9%*
200 ms	16	4.5%*	3.1%	2.3%	2.9%	11.1%*
100 ms	16	4.7%*	3.3%	2.7%	2.7%	n/a
50 ms	16	3.5%*	3.2%	3.1%	2.6%	n/a
1 ms	16	5.9%*	4.5%*	2.9%	5.0%*	n/a
440 ms	8	6.5%*	3.0%	4.1%*	6.3%*	11.1%*
200 ms	8	5.2%*	2.8%	1.9%	2.3%	9.4%*
100 ms	8	3.6%*	3.3%	3.2%	2.7%	n/a
50 ms	8	5.2%*	3.1%	2.3%	3.6%*	n/a
1 ms	8	4.7%*	3.8%*	2.9%	4.2%*	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	16.7%*	10.4%	9.1%	7.0%	69.0%*
2000 ms	96	14.3%*	8.9%	9.1%	8.3%	67.4%*
1000 ms	96	16.1%*	7.8%	10.7%	8.9%	65.9%*
500 ms	96	18.5%*	8.6%	9.4%	8.1%	n/a
1 ms	96	10.9%	10.4%	7.8%	11.2%	n/a
4000 ms	64	12.2%	10.9%	8.3%	8.3%	42.2%*
2000 ms	64	17.7%*	9.1%	7.3%	9.9%	53.4%*
1000 ms	64	14.6%*	9.6%	7.0%	6.8%	55.7%*
500 ms	64	16.7%*	10.4%	7.6%	8.6%	n/a
1 ms	64	9.1%	5.5%	6.3%	7.8%	n/a
4000 ms	32	12.5%*	8.6%	8.1%	8.1%	28.6%*
2000 ms	32	13.0%*	8.3%	6.8%	9.9%	35.9%*
1000 ms	32	13.8%*	8.1%	6.5%	7.3%	43.2%*
500 ms	32	11.5%	9.4%	8.9%	12.5%*	n/a
1 ms	32	8.3%	9.4%	8.1%	9.1%	n/a
4000 ms	24	14.1%*	7.8%	8.3%	7.3%	26.8%*
2000 ms	24	11.5%	8.6%	9.6%	10.9%	28.9%*
1000 ms	24	15.1%*	8.9%	7.6%	7.8%	34.6%*
500 ms	24	14.3%*	9.4%	9.4%	9.4%	n/a
1 ms	24	9.4%	9.6%	9.4%	9.6%	n/a
4000 ms	16	10.7%	8.1%	7.8%	8.6%	25.0%*
2000 ms	16	9.9%	6.2%	8.9%	11.2%	34.6%*
1000 ms	16	13.0%*	9.1%	10.7%	4.7%	31.5%*
500 ms	16	13.0%*	8.6%	9.6%	7.0%	n/a
1 ms	16	7.6%	9.9%	8.3%	8.6%	n/a
4000 ms	8	9.6%	7.3%	6.3%	7.0%	21.9%*
2000 ms	8	11.2%	8.9%	9.1%	6.0%	20.1%*
1000 ms	8	12.0%	8.1%	7.3%	6.5%	25.8%*
500 ms	8	8.9%	7.8%	8.3%	8.6%	n/a
1 ms	8	12.2%	12.0%	7.6%	9.9%	n/a

Tables 11 and 12 in the appendix in the supplementary material, available online, contain data for the other block runs for subject 6 while Tables 21, 22, 23, 24, and 25 and 51, 52, 53, and 54 contain data for all block runs of all other subjects.

TABLE 5
Application of the Analysis From Table 3 to the Rapid-Event Run of Subject 6 on (left) Image and (right) Video Stimuli, Where the Data has been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	2.5%	2.5%	2.4%	3.2%	2.3%
200 ms	96	2.6%	2.4%	2.5%	2.0%	2.5%
100 ms	96	2.5%	1.8%	2.4%	2.9%	n/a
50 ms	96	2.9%	3.1%	3.0%	2.3%	n/a
1 ms	96	3.1%	3.0%	2.1%	2.4%	n/a
440 ms	64	2.7%	2.1%	2.4%	2.8%	2.3%
200 ms	64	1.7%	3.0%	2.5%	2.0%	2.2%
100 ms	64	3.1%	3.0%	2.3%	2.3%	n/a
50 ms	64	3.1%	2.4%	2.3%	2.0%	n/a
1 ms	64	2.5%	2.6%	2.9%	2.0%	n/a
440 ms	32	2.8%	2.3%	1.9%	2.3%	2.8%
200 ms	32	2.5%	2.4%	2.3%	2.1%	2.6%
100 ms	32	2.2%	2.1%	1.8%	2.5%	n/a
50 ms	32	2.5%	2.7%	2.2%	2.3%	n/a
1 ms	32	2.1%	3.0%	3.0%	2.9%	n/a
440 ms	24	2.2%	2.8%	2.3%	2.2%	2.7%
200 ms	24	2.8%	2.9%	2.1%	2.5%	2.7%
100 ms	24	2.5%	2.1%	2.3%	2.3%	n/a
50 ms	24	2.4%	2.6%	2.0%	2.5%	n/a
1 ms	24	2.9%	2.9%	1.8%	2.8%	n/a
440 ms	16	2.2%	2.5%	2.1%	2.2%	3.6%*
200 ms	16	2.2%	3.4%	2.0%	1.9%	2.6%
100 ms	16	2.7%	2.5%	2.7%	2.8%	n/a
50 ms	16	2.7%	3.0%	2.1%	2.9%	n/a
1 ms	16	2.0%	2.5%	2.7%	2.4%	n/a
440 ms	8	2.5%	2.4%	2.7%	3.2%	3.4%
200 ms	8	2.6%	2.8%	2.8%	2.2%	2.7%
100 ms	8	2.2%	2.6%	2.6%	2.4%	n/a
50 ms	8	2.7%	2.5%	2.8%	2.5%	n/a
1 ms	8	2.4%	2.4%	2.3%	2.5%	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	9.4%	6.2%	7.0%	8.1%	12.5%*
2000 ms	96	7.8%	10.2%	6.5%	9.4%	8.6%
1000 ms	96	7.8%	7.0%	7.8%	8.9%	9.4%
500 ms	96	9.9%	7.0%	8.1%	4.9%	n/a
1 ms	96	11.5%	7.0%	9.9%	10.2%	n/a
4000 ms	64	8.9%	7.3%	5.7%	6.8%	10.9%
2000 ms	64	8.9%	7.6%	7.8%	7.6%	9.9%
1000 ms	64	5.5%	7.0%	9.4%	6.5%	9.6%
500 ms	64	8.9%	8.3%	8.1%	7.0%	n/a
1 ms	64	9.4%	9.9%	11.2%	10.4%	n/a
4000 ms	32	8.9%	7.3%	8.3%	8.9%	11.5%
2000 ms	32	9.6%	7.6%	9.4%	9.4%	11.7%
1000 ms	32	7.6%	10.2%	5.7%	9.4%	10.4%
500 ms	32	9.6%	8.3%	7.8%	6.8%	n/a
1 ms	32	7.0%	7.6%	7.6%	9.1%	n/a
4000 ms	24	9.6%	7.6%	6.3%	8.3%	13.0%*
2000 ms	24	7.6%	8.1%	8.9%	9.9%	10.4%
1000 ms	24	8.9%	9.1%	9.4%	8.6%	8.9%
500 ms	24	7.3%	11.2%	10.2%	8.9%	n/a
1 ms	24	7.8%	7.0%	6.5%	8.3%	n/a
4000 ms	16	9.6%	9.4%	7.3%	8.9%	11.2%
2000 ms	16	10.2%	7.3%	6.2%	7.6%	9.4%
1000 ms	16	7.0%	9.1%	6.3%	8.1%	8.3%
500 ms	16	7.8%	8.3%	7.8%	8.6%	n/a
1 ms	16	7.0%	7.0%	7.0%	8.6%	n/a
4000 ms	8	8.3%	7.8%	7.8%	8.6%	11.2%
2000 ms	8	8.3%	8.9%	6.5%	7.8%	6.8%
1000 ms	8	9.9%	9.9%	10.2%	10.2%	10.4%
500 ms	8	8.3%	6.8%	12.8%*	7.8%	n/a
1 ms	8	9.1%	8.9%	9.4%	9.4%	n/a

Tables 26, 27, 28, 29, and 30 in the appendix in the supplementary material, available online, contain data for all other subjects.

given subject and then tested the models on a different block run for that same subject. Since we had three block runs for subject 6, two collected with the same stimulus presentation order and one collected with a different order, this allowed determining both the degree to which classification accuracy observed with a block design depended on having training and test samples from the same block and how much depended on stimulus presentation order. All of these

TABLE 6
Application of the Analysis From Table 3 to the First Block Run of Subject 6 on (left) Image and (right) Video Stimuli, Where the Data has not been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	70.1%*	97.2%*	99.9%*	38.7%*	95.2%*
200 ms	96	71.0%*	97.0%*	99.8%*	49.1%*	95.7%*
100 ms	96	74.5%*	97.2%*	99.8%*	61.4%*	n/a
50 ms	96	72.1%*	97.3%*	99.9%*	77.4%*	n/a
1 ms	96	66.4%*	97.0%*	99.7%*	93.5%*	n/a
440 ms	64	70.0%*	99.1%*	100.0%*	42.2%*	83.1%*
200 ms	64	59.5%*	99.0%*	100.0%*	47.9%*	89.1%*
100 ms	64	71.4%*	98.9%*	99.9%*	59.1%*	n/a
50 ms	64	73.2%*	99.0%*	99.9%*	81.5%*	n/a
1 ms	64	57.4%*	98.6%*	99.7%*	88.4%*	n/a
440 ms	32	55.9%*	98.8%*	100.0%*	37.7%*	79.2%*
200 ms	32	64.0%*	98.8%*	100.0%*	54.1%*	83.1%*
100 ms	32	60.1%*	98.9%*	100.0%*	71.1%*	n/a
50 ms	32	68.5%*	98.6%*	99.9%*	83.6%*	n/a
1 ms	32	55.2%*	98.8%*	98.8%*	68.6%*	n/a
440 ms	24	57.3%*	98.8%*	100.0%*	40.8%*	81.1%*
200 ms	24	58.5%*	98.5%*	99.9%*	57.7%*	80.1%*
100 ms	24	62.8%*	98.7%*	100.0%*	75.7%*	n/a
50 ms	24	64.0%*	98.5%*	100.0%*	85.9%*	n/a
1 ms	24	51.0%*	98.5%*	98.4%*	67.2%*	n/a
440 ms	16	52.5%*	98.9%*	100.0%*	45.2%*	76.5%*
200 ms	16	56.6%*	98.6%*	100.0%*	67.0%*	74.3%*
100 ms	16	62.6%*	98.8%*	99.9%*	77.4%*	n/a
50 ms	16	61.4%*	98.1%*	99.8%*	77.3%*	n/a
1 ms	16	46.7%*	98.0%*	97.2%*	66.1%*	n/a
440 ms	8	42.8%*	98.5%*	99.8%*	66.7%*	78.5%*
200 ms	8	47.5%*	98.5%*	99.8%*	75.1%*	76.9%*
100 ms	8	47.8%*	98.4%*	99.8%*	80.5%*	n/a
50 ms	8	61.0%*	98.4%*	99.7%*	76.1%*	n/a
1 ms	8	49.3%*	97.9%*	95.3%*	64.2%*	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	93.8%*	92.4%*	98.4%*	82.0%*	97.1%*
2000 ms	96	97.9%*	93.2%*	97.9%*	90.4%*	97.7%*
1000 ms	96	96.9%*	91.9%*	97.7%*	92.4%*	96.9%*
500 ms	96	90.9%*	90.6%*	97.1%*	96.1%*	n/a
1 ms	96	90.9%*	91.7%*	98.2%*	97.1%*	n/a
4000 ms	64	91.9%*	93.2%*	98.4%*	76.3%*	97.4%*
2000 ms	64	95.1%*	92.7%*	97.7%*	91.7%*	96.9%*
1000 ms	64	89.3%*	91.9%*	97.9%*	93.8%*	96.6%*
500 ms	64	90.9%*	92.4%*	98.4%*	96.1%*	n/a
1 ms	64	86.2%*	91.1%*	97.7%*	96.6%*	n/a
4000 ms	32	72.9%*	88.0%*	98.4%*	68.2%*	95.8%*
2000 ms	32	78.9%*	89.8%*	97.7%*	84.1%*	91.7%*
1000 ms	32	74.7%*	88.8%*	97.7%*	86.2%*	91.4%*
500 ms	32	79.2%*	89.6%*	97.7%*	91.4%*	n/a
1 ms	32	77.3%*	88.8%*	97.4%*	84.9%*	n/a
4000 ms	24	71.4%*	89.6%*	98.2%*	76.8%*	95.8%*
2000 ms	24	79.7%*	90.4%*	97.9%*	80.2%*	93.2%*
1000 ms	24	67.7%*	89.8%*	98.7%*	89.6%*	94.5%*
500 ms	24	74.0%*	90.1%*	98.2%*	93.2%*	n/a
1 ms	24	63.3%*	89.6%*	96.9%*	87.0%*	n/a
4000 ms	16	63.0%*	89.8%*	98.2%*	74.0%*	92.4%*
2000 ms	16	66.9%*	91.4%*	98.2%*	88.5%*	93.5%*
1000 ms	16	74.5%*	90.4%*	97.7%*	90.1%*	91.9%*
500 ms	16	73.2%*	89.6%*	97.4%*	91.7%*	n/a
1 ms	16	64.8%*	90.1%*	94.0%*	85.4%*	n/a
4000 ms	8	71.6%*	89.1%*	95.3%*	77.9%*	93.2%*
2000 ms	8	58.9%*	88.3%*	94.0%*	87.2%*	89.6%*
1000 ms	8	54.7%*	87.2%*	95.6%*	92.2%*	89.6%*
500 ms	8	53.9%*	87.8%*	94.5%*	89.6%*	n/a
1 ms	8	59.6%*	90.6%*	92.2%*	82.0%*	n/a

Tables 13 and 14 in the Appendix in the supplementary material, available online, contain data for the other block runs for subject 6 while Tables 31, 32, 32, 34, and 35 and 55, 56, 57, and 58 contain data for all block runs of all other subjects.

TABLE 7
Application of the Analysis From Table 3 to the Rapid-Event Run of Subject 6 on (left) Image and (right) Video Stimuli, Where the Data has not been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	1.2%	1.6%	2.9%	1.0%	2.7%
200 ms	96	1.7%	1.8%	3.7%*	0.9%	1.3%
100 ms	96	1.7%	1.9%	3.0%	1.3%	n/a
50 ms	96	1.7%	1.7%	3.3%	1.4%	n/a
1 ms	96	1.2%	1.7%	3.3%	1.1%	n/a
440 ms	64	1.2%	1.6%	3.0%	1.2%	2.3%
200 ms	64	1.4%	1.7%	3.6%*	1.6%	1.3%
100 ms	64	1.4%	1.5%	3.2%	1.4%	n/a
50 ms	64	1.9%	1.8%	3.2%	1.4%	n/a
1 ms	64	1.6%	1.6%	3.2%	1.2%	n/a
440 ms	32	1.5%	1.7%	3.4%	1.3%	2.1%
200 ms	32	1.5%	1.6%	4.2%*	1.1%	1.7%
100 ms	32	1.7%	1.4%	3.3%	1.9%	n/a
50 ms	32	2.1%	1.7%	3.5%	1.1%	n/a
1 ms	32	1.4%	1.7%	4.0%*	1.1%	n/a
440 ms	24	1.4%	1.7%	3.1%	1.8%	1.7%
200 ms	24	1.5%	1.4%	3.7%*	1.2%	1.3%
100 ms	24	1.6%	1.5%	2.8%	1.7%	n/a
50 ms	24	1.8%	1.4%	3.1%	1.5%	n/a
1 ms	24	1.5%	1.2%	3.8%*	1.0%	n/a
440 ms	16	1.8%	1.7%	2.7%	1.2%	1.8%
200 ms	16	1.8%	1.5%	3.1%	1.3%	1.9%
100 ms	16	2.0%	1.5%	3.4%	1.7%	n/a
50 ms	16	2.0%	1.3%	3.2%	1.8%	n/a
1 ms	16	1.7%	1.6%	3.9%*	1.6%	n/a
440 ms	8	1.7%	1.8%	3.7%*	1.5%	2.6%
200 ms	8	1.7%	1.9%	2.7%	1.4%	2.3%
100 ms	8	1.6%	1.8%	2.9%	1.4%	n/a
50 ms	8	1.0%	1.9%	2.8%	1.3%	n/a
1 ms	8	1.4%	1.4%	4.1%*	1.4%	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	9.9%	8.6%	8.3%	9.6%	12.2%
2000 ms	96	8.1%	8.3%	8.6%	10.9%	9.4%
1000 ms	96	6.3%	8.9%	10.2%	10.2%	8.6%
500 ms	96	7.8%	8.1%	7.8%	10.4%	n/a
1 ms	96	8.3%	7.3%	8.6%	10.4%	n/a
4000 ms	64	9.1%	6.8%	9.1%	8.9%	10.4%
2000 ms	64	5.2%	7.3%	7.6%	10.7%	8.9%
1000 ms	64	7.8%	8.9%	6.5%	8.6%	7.8%
500 ms	64	6.8%	8.9%	10.4%	9.1%	n/a
1 ms	64	8.9%	8.1%	7.6%	9.9%	n/a
4000 ms	32	8.9%	7.6%	7.3%	8.1%	9.9%
2000 ms	32	8.1%	8.1%	7.6%	8.9%	8.1%
1000 ms	32	7.3%	9.1%	6.5%	10.7%	5.7%
500 ms	32	9.4%	7.0%	9.9%	7.3%	n/a
1 ms	32	8.3%	11.7%	10.4%	8.3%	n/a
4000 ms	24	7.8%	7.3%	9.4%	7.0%	8.6%
2000 ms	24	7.0%	9.6%	6.3%	10.4%	7.3%
1000 ms	24	9.4%	7.6%	5.5%	8.6%	6.5%
500 ms	24	7.3%	8.1%	6.5%	7.0%	n/a
1 ms	24	7.3%	10.7%	8.6%	10.4%	n/a
4000 ms	16	7.6%	8.1%	8.3%	9.6%	9.1%
2000 ms	16	9.1%	9.4%	10.7%	9.1%	8.1%
1000 ms	16	7.6%	8.6%	11.2%	7.8%	9.1%
500 ms	16	9.1%	9.1%	7.3%	8.6%	n/a
1 ms	16	7.0%	8.1%	6.5%	8.1%	n/a
4000 ms	8	9.9%	8.3%	9.4%	11.2%	9.6%
2000 ms	8	9.1%	7.6%	8.9%	9.1%	9.4%
1000 ms	8	8.1%	5.5%	8.3%	7.8%	7.8%
500 ms	8	6.8%	10.9%	8.1%	7.3%	n/a
1 ms	8	6.0%	6.2%	8.6%	6.3%	n/a

Tables 36, 37, 38, 39, and 40 in the Appendix in the supplementary material, available online, contain data for all other subjects.

analyses between two different block runs average across training on each block run and testing on the other. Table 15 in the Appendix in the supplementary material, available online, illustrates cross-block accuracy when both block runs have the same stimulus presentation order and the data has not been preprocessed with bandpass filtering. Note that classification accuracy drops precipitously from the results in Table 6, though still above chance. Table 18 in

TABLE 8
Reanalysis of the Data Released by OP_1 With Classification Accuracy Averaged Over Leave-One-Subject-Out Round-Robin Cross Validation Instead of the Provided Splits

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	128	2.7%	3.6%*	3.0%*	3.7%*	3.3%*
200 ms	128	2.9%*	3.7%*	2.9%*	3.4%*	3.2%*
100 ms	128	2.8%	3.7%*	2.9%*	3.5%*	n/a
50 ms	128	3.3%*	3.4%*	3.0%*	2.7%	n/a
1 ms	128	2.9%*	2.6%	2.8%	2.7%	n/a
440 ms	96	2.9%	3.2%*	2.9%*	3.1%*	3.4%*
200 ms	96	2.6%	3.1%*	2.9%	3.5%*	3.2%*
100 ms	96	2.3%	2.7%	3.1%*	3.0%*	n/a
50 ms	96	2.2%	2.8%	3.1%*	2.7%	n/a
1 ms	96	2.5%	2.7%	3.0%*	3.2%*	n/a
440 ms	64	2.5%	2.7%	2.5%	3.5%*	3.7%*
200 ms	64	2.7%	2.5%	2.7%	4.2%*	3.7%*
100 ms	64	2.2%	2.9%*	3.0%*	4.0%*	n/a
50 ms	64	2.8%	2.8%	2.5%	3.3%*	n/a
1 ms	64	3.0%*	3.0%*	3.2%*	2.5%	n/a
440 ms	32	2.2%	1.8%	3.2%*	2.9%*	3.0%*
200 ms	32	2.5%	1.8%	3.0%*	3.1%*	2.6%
100 ms	32	2.8%	2.0%	2.8%	3.2%*	n/a
50 ms	32	2.1%	1.9%	3.1%*	3.2%*	n/a
1 ms	32	2.9%*	2.9%	2.8%	2.7%	n/a
440 ms	24	2.1%	2.4%	3.4%*	2.4%	2.6%
200 ms	24	2.8%	2.4%	3.4%*	2.7%	3.1%*
100 ms	24	3.4%*	2.5%	3.3%*	2.9%*	n/a
50 ms	24	3.1%*	2.4%	3.0%*	2.7%	n/a
1 ms	24	3.4%*	2.4%	3.1%*	2.8%	n/a
440 ms	16	2.8%	2.1%	2.6%	2.3%	2.7%
200 ms	16	3.2%*	2.1%	2.6%	2.3%	3.6%*
100 ms	16	3.4%*	1.9%	2.3%	2.7%	n/a
50 ms	16	2.4%	2.0%	2.4%	2.7%	n/a
1 ms	16	2.7%	2.6%	2.8%	2.5%	n/a
440 ms	8	2.1%	2.7%	2.5%	2.9%*	2.1%
200 ms	8	2.0%	2.5%	2.2%	2.8%	2.3%
100 ms	8	2.5%	2.9%*	2.6%	1.9%	n/a
50 ms	8	2.6%	2.6%	2.5%	2.5%	n/a
1 ms	8	2.7%	2.4%	2.5%	2.9%*	n/a

the Appendix in the supplementary material, available online, repeats the analysis from Table 15 in the Appendix in the supplementary material, available online, where the data has been preprocessed with bandpass filtering. Classification accuracy drops to chance. Table 16 in the Appendix in the supplementary material, available online, illustrates cross-block accuracy when the two block runs have different stimulus presentation order but the data has been preprocessed with bandpass filtering. Classification accuracy again drops to chance.

Third, we reran all of the analyses from Table 6 on our new data collected with a rapid-event design, both with and without bandpass filtering, but with a twist. Instead of using correct labels, which varied on a stimulus-by-stimulus basis, we used arbitrary labels, which varied on a block-by-block basis: each block was given a distinct label but all stimuli within a block were given the same label. Thus while the stimuli are changing in each block, they are given the wrong unchanging label and, like the block design employed by OP_1, each trial in the test set comes from a block with many trials in the training set. The results with and without bandpass filtering are shown in Tables 9 and 10 respectively and mirror the results in Tables 4 and 6 respectively. Note that with bandpass filtering, we obtain classification accuracies far higher than chance with the 1D CNN, while without bandpass filtering, we obtain near perfect classification accuracies, similar to those obtained in Tables 1, 2, and 3.

Fourth, we reran the code released by OP_1 (an LSTM combined with a fully connected layer and a ReLU layer) on the data released by OP_1 but first applied various highpass filters with 14 Hz, 10 Hz, and 5 Hz cut-offs to the data. Recall, from Table 1, that we obtain a classification accuracy of 93 percent without such highpass filtering. With the highpass filtering, classification accuracy drops to 32.4 percent (14 Hz), 29.8 percent (10 Hz), and 29.7 percent (5 Hz).

3.8 Regression

In support of claim 2, Spampinato *et al.* [1, Sections 3.3 and 4.2] report an analysis whereby they use the LSTM, combined with a fully connected layer and a ReLU layer, that was trained on EEG data as an encoder to produce a 128-element encoding vector for each image in their dataset. They then regress the 1,000-element output representation from a number of existing deep-learning object classifiers that have been pretrained on ImageNet to produce the same encoding vectors. When training this regressor, in some instances, they freeze the parameters of the existing deep-learning object classifiers, while in other instances they fine tune them while learning the regressor. They report a mean square error (MSE) between 0.62 and 7.63 on the test set depending on the particulars of the model and training regimen [1, Table 4]. They claim that this result supports the conclusion that this is *the first human brain–driven automated visual classification method* and thus enables *automated visual classification in a "brain-based visual object manifold"* [1, Section 5 p. 6816].

Note that OP_1 use the same LSTM combined with a fully connected layer and a ReLU layer both as a classifier and as an encoder. During training as a classifier, the output of the last layer of the classifier, namely the ReLU, is trained to match the class label. Thus using such a trained classifier as an encoder would tend to encode EEG data in a representation that is close to class labels. Crucially, the output of the classifier taken as an encoder contains mostly, if not exclusively, class information and little or no reflection of other non-class-related visual information. Further, since the output of their classifier is a 128-element vector, since they have 40 classes, and since they train with a cross-entropy loss that combines log softmax with a negative log likelihood loss, the classifier tends to produce an output representation whose first 40 elements contain an approximately one-hot-encoded representation of the class label, leaving the remaining elements at zero. Indeed, we observe this property of the encodings produced by the code released by OP_1 on the data released by OP_1 (Fig. 3 in the Appendix in the supplementary material), available online. Note that the diagonal nature of Fig. 3 in the Appendix in the supplementary material, available online, reflects an approximate one-hot class encoding. Any use of a classifier trained in this fashion as an encoder would have this property. Spampinato *et al.* [1, Sections 3.3, 4.2, and 4.4] use such an encoder to train an object classifier with EEG data, Palazzo *et al.* [3], Kavasidis *et al.* [4], and Tirupattur *et al.* [7] use such an encoder to train a variational autoencoder (VAE) [30] or a generative adversarial network (GAN) [31] to produce images of human perception and thought, and

TABLE 9
Reanalysis of the Rapid-Event Run for Subject 6 on (left) Image and (right) Video Stimuli With Incorrect Block-Level Labels, Where the Data has Been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	22.7%*	5.9%*	11.3%*	9.0%*	59.6%*
200 ms	96	15.1%*	5.3%*	3.0%	2.2%	54.3%*
100 ms	96	15.7%*	4.2%*	3.4%	3.6%*	n/a
50 ms	96	13.0%*	5.0%*	3.1%	2.8%	n/a
1 ms	96	7.5%*	4.6%*	2.8%	5.4%*	n/a
440 ms	64	21.5%*	6.2%*	11.7%*	8.2%*	48.2%*
200 ms	64	9.7%*	3.7%*	2.9%	2.4%	40.7%*
100 ms	64	5.8%*	3.3%	2.8%	2.8%	n/a
50 ms	64	5.9%*	2.7%	3.2%	2.7%	n/a
1 ms	64	5.4%*	4.0%*	2.7%	3.5%*	n/a
440 ms	32	16.5%*	7.2%*	13.3%*	9.3%*	37.7%*
200 ms	32	4.9%*	2.9%	1.9%	2.7%	22.2%*
100 ms	32	4.8%*	3.6%*	2.5%	2.6%	n/a
50 ms	32	3.8%*	2.8%	3.0%	3.1%	n/a
1 ms	32	4.8%*	3.8%*	2.1%	3.8%*	n/a
440 ms	24	16.8%*	8.2%*	14.8%*	8.9%*	34.8%*
200 ms	24	4.3%*	2.9%	2.4%	2.9%	18.4%*
100 ms	24	3.9%*	3.2%	2.5%	2.5%	n/a
50 ms	24	3.8%*	3.4%	2.3%	2.0%	n/a
1 ms	24	5.9%*	5.0%*	2.5%	4.5%*	n/a
440 ms	16	16.9%*	9.8%*	15.0%*	10.0%*	31.3%*
200 ms	16	4.5%*	3.3%	2.9%	3.2%	11.0%*
100 ms	16	3.5%*	3.1%	3.1%	2.6%	n/a
50 ms	16	3.4%	2.8%	2.5%	2.4%	n/a
1 ms	16	5.1%*	3.6%*	2.8%	4.0%*	n/a
440 ms	8	16.2%*	11.8%*	16.4%*	11.1%*	28.9%*
200 ms	8	3.6%*	3.3%	2.5%	3.1%	7.5%*
100 ms	8	3.4%	2.7%	1.8%	2.7%	n/a
50 ms	8	3.2%	2.6%	2.7%	2.6%	n/a
1 ms	8	5.3%*	4.6%*	3.1%	4.2%*	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	26.0%*	10.7%	14.6%*	12.5%*	70.3%*
2000 ms	96	24.0%*	8.9%	9.1%	9.6%	72.4%*
1000 ms	96	21.4%*	9.6%	6.8%	7.8%	65.9%*
500 ms	96	21.4%*	10.7%	10.7%	10.4%	n/a
1 ms	96	10.2%	8.6%	8.6%	9.1%	n/a
4000 ms	64	19.8%*	9.1%	10.9%	10.9%	54.7%*
2000 ms	64	18.0%*	8.1%	7.3%	7.0%	49.0%*
1000 ms	64	17.2%*	7.6%	6.3%	8.9%	51.0%*
500 ms	64	21.9%*	7.0%	9.1%	6.5%	n/a
1 ms	64	8.9%	11.2%	8.1%	9.4%	n/a
4000 ms	32	18.2%*	8.3%	12.5%*	11.2%	38.0%*
2000 ms	32	16.9%*	6.5%	9.9%	10.9%	40.1%*
1000 ms	32	13.5%*	10.4%	8.3%	5.5%	32.6%*
500 ms	32	17.2%*	7.8%	10.2%	12.8%*	n/a
1 ms	32	12.0%	9.4%	7.0%	9.4%	n/a
4000 ms	24	18.0%*	8.6%	12.2%	10.7%	39.6%*
2000 ms	24	15.6%*	8.1%	7.8%	9.6%	33.6%*
1000 ms	24	14.1%*	9.4%	5.5%	6.5%	39.6%*
500 ms	24	12.8%*	7.0%	8.1%	8.9%	n/a
1 ms	24	10.2%	12.5%*	9.6%	9.4%	n/a
4000 ms	16	15.4%*	8.3%	13.0%*	11.5%	38.3%*
2000 ms	16	11.5%	9.4%	6.8%	7.0%	26.6%*
1000 ms	16	12.0%	9.6%	8.9%	9.1%	25.3%*
500 ms	16	17.2%*	8.9%	7.6%	6.5%	n/a
1 ms	16	11.5%	11.5%	11.2%	10.7%	n/a
4000 ms	8	17.2%*	8.3%	14.8%*	13.5%*	35.7%*
2000 ms	8	15.6%*	8.9%	9.1%	8.3%	27.1%*
1000 ms	8	10.2%	9.1%	6.3%	9.6%	18.8%*
500 ms	8	10.9%	9.9%	10.2%	9.1%	n/a
1 ms	8	8.9%	10.2%	7.6%	8.3%	n/a

Tables 41, 42, 43, 44, and 45 in the Appendix in the supplementary material, available online, contain data for all other subjects.

TABLE 10
Reanalysis of the Rapid-Event Run for Subject 6 on (left) Image and (right) Video Stimuli With Incorrect Block-Level Labels, Where the Data has not been Preprocessed With Bandpass Filtering

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
440 ms	96	95.2%*	99.2%*	100.0%*	89.4%*	100.0%*
200 ms	96	91.2%*	99.0%*	100.0%*	95.7%*	99.9%*
100 ms	96	92.4%*	99.1%*	100.0%*	98.6%*	n/a
50 ms	96	95.5%*	99.2%*	100.0%*	98.5%*	n/a
1 ms	96	80.8%*	98.7%*	100.0%*	99.7%*	n/a
440 ms	64	87.7%*	99.4%*	100.0%*	91.0%*	99.8%*
200 ms	64	80.1%*	99.4%*	100.0%*	94.9%*	99.8%*
100 ms	64	87.3%*	99.5%*	100.0%*	98.5%*	n/a
50 ms	64	86.2%*	98.9%*	99.9%*	99.6%*	n/a
1 ms	64	64.3%*	98.9%*	99.9%*	99.7%*	n/a
440 ms	32	81.1%*	98.6%*	99.9%*	84.6%*	99.3%*
200 ms	32	72.4%*	98.7%*	99.7%*	93.3%*	97.9%*
100 ms	32	79.3%*	98.9%*	99.9%*	97.2%*	n/a
50 ms	32	77.3%*	98.7%*	99.8%*	96.9%*	n/a
1 ms	32	61.1%*	97.3%*	99.4%*	97.3%*	n/a
440 ms	24	79.7%*	98.9%*	99.9%*	89.5%*	98.9%*
200 ms	24	72.4%*	99.1%*	99.8%*	93.9%*	98.3%*
100 ms	24	75.4%*	98.8%*	99.9%*	96.9%*	n/a
50 ms	24	77.7%*	98.9%*	99.7%*	98.6%*	n/a
1 ms	24	60.1%*	97.7%*	99.4%*	96.3%*	n/a
440 ms	16	68.3%*	98.9%*	99.7%*	91.6%*	97.4%*
200 ms	16	65.8%*	98.8%*	99.7%*	95.1%*	97.9%*
100 ms	16	65.0%*	98.6%*	99.8%*	97.7%*	n/a
50 ms	16	70.0%*	98.6%*	99.5%*	98.4%*	n/a
1 ms	16	56.3%*	97.4%*	99.1%*	94.3%*	n/a
440 ms	8	50.0%*	97.8%*	98.6%*	90.0%*	93.7%*
200 ms	8	48.6%*	97.8%*	98.5%*	94.5%*	91.6%*
100 ms	8	59.4%*	97.7%*	98.2%*	95.8%*	n/a
50 ms	8	70.7%*	96.7%*	98.3%*	95.7%*	n/a
1 ms	8	53.5%*	95.0%*	96.2%*	84.1%*	n/a

window	channels	LSTM	k-NN	SVM	MLP	1D CNN
4000 ms	96	97.4%*	89.1%*	97.1%*	89.8%*	97.1%*
2000 ms	96	97.1%*	89.1%*	98.2%*	96.4%*	97.4%*
1000 ms	96	96.1%*	90.9%*	97.4%*	97.4%*	97.1%*
500 ms	96	97.4%*	89.8%*	98.2%*	98.2%*	n/a
1 ms	96	89.3%*	87.8%*	96.6%*	96.9%*	n/a
4000 ms	64	84.4%*	89.6%*	97.4%*	86.5%*	96.6%*
2000 ms	64	92.7%*	90.9%*	97.4%*	91.7%*	97.1%*
1000 ms	64	94.8%*	89.3%*	97.7%*	94.8%*	94.3%*
500 ms	64	89.1%*	89.6%*	96.9%*	97.4%*	n/a
1 ms	64	91.9%*	88.3%*	97.4%*	97.4%*	n/a
4000 ms	32	86.7%*	89.1%*	98.2%*	87.2%*	95.8%*
2000 ms	32	88.3%*	90.6%*	98.4%*	93.8%*	94.3%*
1000 ms	32	90.6%*	89.8%*	98.2%*	95.8%*	96.4%*
500 ms	32	87.0%*	91.1%*	98.7%*	97.7%*	n/a
1 ms	32	76.6%*	89.6%*	97.1%*	92.7%*	n/a
4000 ms	24	84.6%*	89.6%*	98.2%*	85.4%*	95.8%*
2000 ms	24	85.4%*	89.6%*	97.9%*	91.1%*	94.5%*
1000 ms	24	84.9%*	91.7%*	98.2%*	96.1%*	94.3%*
500 ms	24	80.2%*	91.1%*	96.6%*	96.1%*	n/a
1 ms	24	77.3%*	91.7%*	97.4%*	90.4%*	n/a
4000 ms	16	82.6%*	92.7%*	98.4%*	89.1%*	96.6%*
2000 ms	16	79.2%*	92.4%*	98.2%*	92.2%*	95.6%*
1000 ms	16	71.4%*	91.9%*	97.9%*	97.7%*	95.1%*
500 ms	16	79.7%*	90.6%*	96.9%*	95.6%*	n/a
1 ms	16	70.6%*	91.4%*	96.1%*	89.8%*	n/a
4000 ms	8	61.2%*	94.5%*	97.9%*	86.7%*	93.2%*
2000 ms	8	62.2%*	93.8%*	98.2%*	90.1%*	93.0%*
1000 ms	8	68.5%*	94.3%*	97.4%*	92.2%*	92.4%*
500 ms	8	74.5%*	92.2%*	97.7%*	91.9%*	n/a
1 ms	8	64.1%*	92.7%*	96.6%*	85.9%*	n/a

Tables 46, 47, 48, 49, and 50 in the Appendix in the supplementary material, available online, contain data for all other subjects.

Palazzo *et al.* [8] use such an encoder to produce saliency maps, EEG activation maps, and to measure association between EEG activity and layers in an object detector. Thus all this work is essentially driven by encodings of class information that lack any visual information or any representation of brain processing.

We ask whether there is merit in the regression algorithm proposed by OP_1 to create a novel object classifier driven by brain signals. We analyze their algorithm under the assumption that it is applied to EEG data that supports classification of visually perceived objects and does not suffer from contamination. Under this assumption, the EEG response of two

images of the same class would be closer than for two images of different classes. An encoder like the one employed by OP_1 would produce encodings that are more similar for images of the same class than images of different classes. (For their actual encoder, Fig. 3 in the Appendix in the supplementary material, available online, shows that they are indeed little more than class encodings.) Moreover, deep-learning object classifiers presumably produce closer representations for images in the same object class than for images of different classes. After all, that is what object classifiers do. Thus all the regressor does is preserve the property that two images of the same class regress to closer representations than two images of different classes. In other words, all the regressor does is map a 1,000-dimension representation of class to a 128-dimension representation of class. It should not matter whether the actual target representation is a reflection of brain processing or not.

We asked whether the putative success of this regression analysis depended on a representation derived from neuroimaging. To this end, we generated a random codebook with random codewords that simulate the EEG response of all six subjects to all 2,000 image stimuli. This was done with the following procedure. We first generated 40 random codewords, one for each class, by uniformly sampling elements i.i.d. in [0, 2]. We then generated $50 \times 6 = 300$ random codewords for each class, one for each subject and image, by adding univariate Gaussian noise with $\sigma^2 = 4$ i.i.d. to the elements of the class codewords, and clipped the elements to be nonnegative. This generated a codebook of 12,000 random codewords for each simulated subject response that has the property that encodings for images in the same class are closer than entries for images in different classes. These codewords carry no brain-inspired meaning whatsoever. Like OP_1, we then averaged the codewords across subject for each image. We then applied the PyTorch VGG-16 [32] pretrained on ImageNet, without any fine tuning, to each of the images in the OP_1 dataset. Finally, we trained a linear regressor with MSE loss and L2 regularization from the output of VGG-16 on each image to the average random codeword for that image on the training set for the first split provided by OP_1. We then measured an average MSE of 0.55 on the validation and test sets of that split. The fact that it is possible to regress the output of an off-the-shelf pretrained object classifier to random class encodings as well as one can regress that output to class encodings derived from an EEG encoder demonstrates that the ability to do so does not depend on anything other than class information in the source and target representations.

3.9 Transfer Learning

In further support of claim 2, Spampinato *et al.* [1, Section 4.4] report an analysis that purports to demonstrate that the learned combination of regressor and object classifier generalizes to other datasets with disjoint sets of classes. To this end, they first apply VGG-16, pretrained on ImageNet, to a subset of the Caltech 101 [33] dataset with 30 classes, not fine tuned, to produce a 1,000-element representation of each image. They then map this with their regressor trained as described above to 128-element encodings. Finally, they train and test an SVM classifier on the resulting encodings. They compare this with an SVM classifier trained and tested on the 1,000-element outputs from pretrained deep-learning object classifiers that have not been mapped with their regressor and achieve comparable performance (92.6 percent on the 1,000-element output of GoogLeNet [36] and 89.7 percent on the 128-element encodings regressed from GoogLeNet). They claim that their approach enables *automated visual classification in a "brain-based visual object manifold"* and *show [s] competitive performance, especially as concerns learning EEG representation of object classes* [1, Section 5 p. 6816].

We conjecture that the putative success of this transfer-learning analysis is not surprising and demonstrates nothing about the quality of the representation nor whether it reflects brain processing. As discussed above, the deep-learning object classifiers produce closer output representations for images in the same object class than for images of different classes. Further, as discussed above, all the regressor does is preserve the property that two images of the same class regress to closer encodings than two images of different classes. The choice of regressor or regressed representation should have no impact on the SVM classifier so long as these properties hold.

We thus asked whether the putative success of this transfer-learning analysis depended on a representation derived from neuroimaging. To this end, we used VGG-16, pretrained on ImageNet without any fine tuning, to map the images in Caltech 101 to 1,000-element encodings and applied the regressor that we trained on random representations to map these 1000-element encodings to 128-element encodings. This composite mapping exhibited the above properties. This, again, generated a codebook of random codewords for each image in this subset of Caltech 101 that has the property that entries for images in the same class are closer than entries for images in different classes. As before, the codewords carry no brain-inspired meaning. We split our subset of Caltech 101 into disjoint training and test sets, trained a linear SVM on the training set, and achieved an accuracy of 95.9 percent on the test set when classified on the 128-element encodings regressed from VGG-16 as compared with 94.9 percent on the test set when classified on the 1,000-element output of VGG-16.

4 Reconciling Discrepancies

A number of papers, e.g., Spampinato *et al.* [2, Figs. 1 and 2 (c)], Palazzo *et al.* [3, Figs. 1 and 2], and Kavasidis *et al.* [4, Figs. 2, 3, and 4], use an encoder that appears to be similar or identical to that reported in Spampinato *et al.* [1, Figs. 2 and 3(c)].[10] A number of papers [3], [4], [5], [8] use the dataset reported in OP_1.[11] OP_1 have released the code[2] for their encoder as well as their data.[3] They have released their data in two formats, Python and Matlab. We have observed a number of discrepancies between the different published accounts, between the different released variants of the data, and between the published accounts and the released

10. Palazzo *et al.* [3] and Tirupattur *et al.* [7] appear to employ related but somewhat different encoders. We do not comment on these here since we do not have access to this code.

11. An earlier but similar dataset was reported in Spampinato *et al.* [2]. Tirupattur *et al.* [7] use a different dataset reported by Kumar *et al.* [6]. We do not comment on these here since we do not have access to these datasets.

code and data. We discuss here how we reconciled such for the purposes of the experiments and analyses reported here. We do this solely to document precisely what we have done. We do not believe that anything substantive turns on these issues, except for the issue of filtering, whether or not the DC and VLF components are removed from the EEG data. In the case of filtering, we perform all analyses twice, with and without such removal. In the case of differences between the published accounts and released code, we have repeated all analyses with both the released code and all reasonable interpretations of the published accounts and observed no substantive difference. The Appendix in the supplementary material, available online, reports all these repeated analyses.

4.1 Filtering

Spampinato *et al.* [2, Section 3.1 p. 7], Spampinato *et al.* [1, Section 3.1 p. 6812], and Palazzo *et al.* [3, Section 3.1 p. 3412] claim to preprocess the EEG data with a bandpass filter (14–70 Hz) and a notch filter (49–51 Hz). Later publications [4], [7], [8] do not discuss filtering. The code originally released by OP_1 does not contain any bandpass or notch filtering, but does contain z-scoring. Further, spectral analysis of their released data suggested that no bandpass filtering was performed. We provided an early draft of this paper to the authors of OP_1 and engaged the authors in email correspondence to clarify the experimental procedure. That correspondence states that

- the original paper(s) did not accurately describe preprocessing,
- the released dataset comes directly from the recording device and was not preprocessed or filtered,
- notch filtering and z-scoring was performed but no other preprocessing was performed,
- this filtering was done during training and is not reflected in the released dataset, and
- all of the reported results were produced with the released code except that the released code lacks the notch filtering which was performed by other nonreleased code.

The authors have subsequently modified their released code to include bandpass and notch filtering. We take our correspondence with the authors to imply that no filtering was applied during acquisition, no filtering was applied prior to production of either the Python or Matlab format released data, the analyses reported in OP_1 were performed using the original released code which did not perform any filtering, and any filtering code was added subsequent to our contact with OP_1. All analyses reported here were performed with the original released code, modified as discussed below, on the Python format data, unmodified, except as discussed below and in the text.

4.2 Quantization

Spampinato *et al.* [1, Section 3.1 p. 6813] and Palazzo *et al.* [3, Section 3.1 p. 3413] report that the EEG data was quantized. As the released code contains no indication of such, we have no way of knowing sufficient details of how to replicate this quantization on our data. We further have no way of knowing if the released Python and/or Matlab data reflects this quantization or not. Thus we do not perform any quantization on either the released data or our new data as part of any analyses reported here.

4.3 Trials Considered

OP_1 nominally collected 50 trials for each of 40 stimuli and 6 subjects for a total of 12,000 trials. However, Palazzo *et al.* [3, Section 3.1 p. 3413] and Kavasidis *et al.* [4, Section 3.1 p. 1811] report that certain trials were discarded. The released data in Python format contains 11,965 trials which is a superset of the released data in Matlab format that contains 11,466 trials. The 499 trials in the Python format data that are missing from the Matlab format data come from Subject 2. Further, the Python format data differs from the Matlab format data. We have no way of knowing why the data differs and why the Python and Matlab format data contain different numbers of trials. Nonetheless, we use all 11,965 trials in the Python format data, including the 499 trials that are missing in the Matlab format data.

4.4 Trial Window

Spampinato *et al.* [1, Section 3.1 p. 6813], Palazzo *et al.* [3, Section 3.1 p. 3413], and Palazzo *et al.* [8, Section 7.1 p. 7] report that samples 40–480 were used. Palazzo *et al.* [3, Section 3.1 p. 3413] report that trials shorter than 480 samples were discarded, those with between 480 and 500 samples were padded with zeros to be 500 samples long, and trials longer than 500 samples were tail trimmed. The released code, however, uses samples 20–450 (i.e., a sequence of length 430), lacks zero padding and tail trimming, and discards sequences shorter than 450 samples or longer than 600 samples. No trials are shorter than 480 samples so none are discarded for this reason and none require zero padding. The released code, however, discards 25 trials beyond the 534 mentioned above for being longer than 600 samples. We have no way of knowing what was actually done to obtain the results in OP_1, Palazzo *et al.* [3], Kavasidis *et al.* [4], and Palazzo *et al.* [8]. Here, we modified the released code to not discard (the 25) trials longer than 600 samples and to use samples 40–480 from each trial instead of 20–450.

4.5 The Encoder Model

When describing the encoder model ([2, Figs. 1 and 2(c)], [3, Figs. 1 and 2], and [4, Figs. 2, 3, and 4]), Spampinato *et al.* [2, Section 3.2 p. 9], Spampinato *et al.* [1, Section 3.2 p. 6813], Palazzo *et al.* [3, Section 3.2 p. 3414], and Kavasidis *et al.* [4, Section 3.2 p. 1811] state that the LSTM layer was followed by a fully connected layer and then a ReLU layer. However, the released code omits the ReLU layer. We modified the released code to add the ReLU layer for the analyses reported here.

4.6 The Classifier

Spampinato *et al.* [2, Fig. 1] and Spampinato *et al.* [1, Fig. 2] report training the encoder by attaching a classifier to its output and training against known labels. Spampinato *et al.* [2, Section 3.2 p. 8, Section 3.3 p. 10, Section 4.3 p. 14], Spampinato *et al.* [1, Section 3.2 p. 6813, Section 3.3 p. 6814, Section 4.3 p. 6816], Palazzo *et al.* [3, Section 3.2 p. 3414], Kavasidis *et al.* [4, Section 3.2 p. 1811, Section 4.3 p. 1815], Tirupattur *et al.* [7, Section 4.3 p. 954], and Palazzo *et al.* [8,

Section 7.2 p. 8] describe this (40-way) classifier alternately as a softmax layer, a softmax classification layer, a softmax classifier, and softmax. Colloquial usage varies as to whether or not this implies use of a fully connected layer prior to the softmax layer.

The released code appears to use PyTorch `torch.nn.functional.cross_entropy`, which internally uses `torch.nn.functional.log_softmax` directly applied to the 128-element output of the encoder, without an intervening fully connected layer. Training a 40-way classifier this way, appended to an encoder, with an implicit one-hot representation of class labels, will tend to train the encoder to produce 128-element EEG encodings where all but the first 40 elements are zero (Fig. 3 in the Appendix in the supplementary material, available online). Indeed, we have observed this behavior with the released code. We have no way of knowing what was actually intended and used to generate the reported results. Here, like the released code, we train the encoders with the same cross-entropy loss, which internally contains a log softmax operation, but use the output of the encoder, prior to any softmax operation, for classification. (Note that had the output of the softmax layer been taken as the EEG encodings, they would have been one-hot.) Nothing turns on this. In the Appendix in the supplementary material, available online, we perform all analyses with both the original unmodified code and four variants that cover all possible reasonable interpretations of what was reported in the published papers. All exhibit the same broad pattern of results.

5 Discussion

The analyses in Section 3.3 demonstrate that the results reported by OP_1 do not depend on the within-stimulus temporal or spatial structure of the EEG signal. In particular, the fact that both the data released by OP_1 and our new data collected with a block design can be classified with extremely short temporal windows demonstrates that the classification performance does not depend on the temporal nature of brain processing. This is exacerbated by the fact that the temporal position of the window can vary randomly between samples in both the training and test sets. The fact that both datasets can be classified with an extremely small number of channels demonstrates that the classification performance does not depend on the spatial nature of brain processing. The new data collection effort in Section 3.4 demonstrates that the results reported by OP_1 crucially depend on their experimental protocol, which can be easily replicated by others, and not on any unique aspect of their data collection effort and laboratory facilities. The analyses in Section 3.5 demonstrate that the results reported by OP_1 crucially depend on a block design and cannot be replicated with a rapid-event design. The analyses in Section 3.7 demonstrate that the results reported by OP_1 crucially depend on contaminated data. The block design of OP_1, together with their training/test-set splits, leads to data contamination, because every trial in each test set comes from a block that has many trials in the corresponding training set.

The first analysis in Section 3.7 shows that if one adopts splits that separate trials from a block so that the test sets never contain trials from blocks that have any trials in the corresponding training sets, classification accuracy drops to chance. This strongly suggests that the high classification accuracy obtained by OP_1 crucially depends on such contamination, which constituted classifying arbitrary temporal artifacts of the data instead of stimulus-related activity.

The second analysis in Section 3.7 shows that the results reported by OP_1 crucially depend not only on a block design, but on shared stimulus-class presentation order. This, together with the fact that the accuracy degrades to chance when the data is preprocessed by bandpass filtering, strongly suggests that even a cross-block analysis of data collected with a block design is classifying long-term temporal characteristics of the EEG signal, not short-term perceptual characteristics of the stimuli. Further, the severe accuracy degradation between a within-block analysis and a cross-block analysis strongly suggests that the high classification accuracy obtained by OP_1 crucially depends on data contamination, which constituted classifying arbitrary temporal artifacts of the data instead of stimulus-related activity.

This is further corroborated by the third analysis in Section 3.7 that shows that one can obtain near perfect classification accuracy with an experiment design where labels vary only by block but where the class of the stimuli within the block are uncorrelated with the labels. If the methods of OP_1 were indeed classifying brain activity due to perception of the class of the stimuli, one would expect to obtain chance performance with this analysis. The fact that near perfect performance was obtained strongly suggests that these methods are indeed classifying the long-term static brain activity that persists during a block that is uncorrelated with the perceptual activity.

Finally, the fourth analysis in Section 3.7 shows that this finding is exacerbated by the presence of DC and VLF components of the recorded EEG signal that are present due to the omission of bandpass filtering.

Simply stated, any EEG experiment with a block design will be contaminated if

- the test set contains trials collected in the same block or in close temporal proximity to trials of the same class in the training set or
- the training and test sets were collected with the same stimulus-class presentation order.

The data collected, used, and released by OP_1 irreparably suffers from contamination. It is impossible to remove the inherent data contamination that results from the fact that every trial in their test sets comes from the same block as many trials in the corresponding training set. This is a property of their experiment design, the block design combined with their training and test set splits. Since only a single block was recorded from each subject, it is impossible to construct splits that decontaminate their data. It is further impossible to decontaminate their data because all block runs were recorded with the same stimulus-class presentation order. Since the data released by OP_1 irreparably suffers from contamination, it renders this dataset unsuitable for its intended purpose of decoding perceptual and conceptual processing and further invalidates all subsequent analyses and claims that use this data for those purposes [3], [4], [5], [8]. We propose that all future classification experiments performed on EEG data employ a design that controls for such contamination.

5.1 Consequences of Flawed Filtering

While OP_1 and two related papers [2], [3] suggest that the reported results were obtained with a process that included bandpass and notch filtering, subsequent analysis and communication with the authors now suggest that this was not the case (Section 4.1). This analysis and communication with the authors has led them to modify their code (Section 4.1). This is important for two reasons. First, no amount of filtering can remove the inherent contamination in their data or the inherent flaws in their experiment design. Second, the fact that the authors omitted the bandpass filter exacerbated the issue, leading to egregious overestimation of the classification accuracy. This has led to their results and data receiving considerable attention and enthusiasm, possibly contributing to the sheer number of papers that use this dataset and/or pursue similar approaches. Had the stated filtering been performed, perhaps the resulting more modest (but still invalid) results would have tempered the rapid proliferation of follow-up work that also suffers from similar methodological shortcomings. We stress that the root issue is data contamination. Lack of filtering is not the root issue; it only exacerbates the root issue.

5.2 Consequences of Flawed Block Design on Subsequent Papers

The above strongly suggests that the output of the LSTM-based encoder trained by OP_1 does not constitute a *"brain-based visual object manifold"* [1, Section 5 p. 6816]. Further, the analyses in Sections 3.8 and 3.9 strongly suggest that the object classifiers constructed by Spampinato *et al.* [1, Sections 3.3, 4.2, and 4.3] are not making use of *any* information in the output of the trained LSTM-based encoder, whether or not it contains a representation of human brain processing. Since these flaws are orthogonal to those of the data contamination issue, these methods are irreparably flawed and their shortcomings would not be remedied by correction of the contamination issue.

Kumar *et al.* [6] report a different EEG dataset that also appears to have been collected with a block design. Data was recorded from a single 10 s block for a single stimulus from each of 30 classes for each of 23 subjects. Each 10 s block was divided into either 40 or 200 segments. Ten-way cross validation was performed during analysis. We have no way of knowing whether the test sets contained segments from the same blocks that had segments in the corresponding training sets. But since a single block was recorded for each stimulus for each subject, the only way to avoid such would have been to conduct cross-subject analyses. The first analysis in Section 3.7 suggests that such cross-subject EEG analysis is difficult and far beyond the current state of the art.

Tirupattur *et al.* [7] report using the dataset from Kumar *et al.* [6] to drive a generative adversarial network (GAN) in a fashion similar to Palazzo *et al.* [3]. That work performs five-way cross validation during analysis. Again, we have no way of knowing whether the test sets contained segments from the same blocks that had segments in the corresponding training sets, and avoiding such would have required cross-subject analyses that our experiments suggest are far beyond the current state of the art.

5.3 Consequences of Using Flawed EEG Encodings as Input to Image Synthesis

Palazzo *et al.* [3], Kavasidis *et al.* [4], and Tirupattur *et al.* [7] all purport to use the EEG encodings to generate images using a GAN that depict human perception and thought. Since we lack access to the code for any of these papers, we are unable to perform the kind of random data analysis that we perform in Sections 3.8 and 3.9 to evaluate these methods. Instead, here we analyze the result in Tirupattur *et al.* [7], using only the published synthesized images. We select this paper because it has the most extensive set of generated examples. Tirupattur *et al.* [7, abstract p. 950] state:

> *While extracting spatio-temporal cues from brain signals for classifying state of human mind is an explored path, decoding and visualizing brain states is new and futuristic. Following this latter direction, in this paper, we propose an approach that is able not only to read the mind, but also to decode and visualize human thoughts. More specifically, we analyze brain activity, recorded by an ElectroEncephaloGram (EEG), of a subject while thinking about a digit, character or an object and synthesize visually the thought item. To accomplish this, we leverage the recent progress of adversarial learning by devising a conditional Generative Adversarial Network (GAN), which takes, as input, encoded EEG signals and generates corresponding images.*

Tirupattur *et al.* [7, Section 1 p. 950] further state:

> *Our goal is to extract some cues from the brain activity, recorded using low-cost EEG devices*[1]*, and use them to visualize the thoughts of a person. More specifically, we attempt to visualize the thoughts of a person by generating an image of an object that the person is thinking about. EEG data of the person is captured while he is thinking of that object and is used for image generation. We use a publicly available EEG dataset [16] for our experiments and propose a generative adversarial model for image generation. We make the following contributions in this work: 1) we introduce the problem of interpreting and visualizing human thoughts, 2) we propose a novel conditional GAN architecture, which generates class-specific images according to specific brain activities; 3) finally, we also show that our proposed GAN architecture is well suited for small-sized datasets and can generate class-specific images even when trained on limited training data.*
>
> *We demonstrate the feasibility and the effectiveness of the proposed method on three different object categories, i.e., digits, characters, and photo objects, and show that our proposed method is, indeed, capable of reading and visualizing human thoughts.*

Conditional GANs are not intended to output exact copies of the training set because the input that leads to synthesized images contains noise in addition to class information. GANs in their true spirit are supposed to learn visual features that are indicative of different instances of objects within a class and synthesize novel images for instances of a class by selecting and combining those features in a semantically and visually coherent fashion. However, essentially all of the example images illustrated in Tirupattur *et al.* [7, Fig. 6] are nearly exact copies of images in ImageNet (Fig. 2). This indicates mode collapse. Moreover, in order for them to

generate the same image twice, they must be provided with the same conditioning input, which in this case comprises both an EEG encoding and noise. It would be highly unlikely for the same EEG encoding and the same noise to be provided at each training iteration. Thus it would be highly unlikely for a proper conditional GAN to be able to memorize the training set. Moreover, it would be highly unlikely for the same EEG encoding and the same noise to be provided both during training and test. Thus it would be highly unlikely for a proper conditional GAN to output near exact copies of the training set during test. Without their code and data, it is impossible for us to precisely determine the cause of this highly unlikely circumstance. Nonetheless, this calls into question their claim that their *proposed method is, indeed, capable of reading and visualizing human thoughts*.

5.4 Consequences for Flawed Joint Training of EEG and Image Encoders to Analyze Brain Processing of Images

Palazzo *et al.* [8, Fig. 1] jointly train an EEG encoder and an image encoder to produce similar encoded representations and then purport to use the trained encoders for several purposes: producing saliency maps [8, Sections 4 and 7.3, and Figs. 3 and 5], producing EEG activation maps [8, Sections 5 and 7.4, and Fig. 6], and associating EEG activity with layers in a synthetic object detector [8, Sections 6 and 7.4, and Fig. 9]. Since these results were all produced with the same contaminated dataset, these results are all suspect. Moreover, Tables 5 and 7 suggest that using the proposed methods to produce legitimate results from uncontaminated data collected with a rapid-event design is unlikely to succeed. Beyond this, however, the methods themselves appear to be fundamentally flawed and unlikely to demonstrate anything, even if they could be made to work on uncontaminated data. The loss function employed in the joint training regimen simply constrains the two encoded representations to be similar. A perfectly trained image encoder, trained against class labels, would simply encode image class, no more and no less. A perfectly trained EEG encoder, trained against class labels, would simply encode stimulus class, no more and no less. During joint training of the EEG encoder, the image encoder serves simply as a surrogate for class labels, no more and no less. Similarly during joint training of the image encoder, the EEG encoder serves simply as a surrogate for class labels, no more and no less. Thus joint training accomplishes nothing that could not be accomplished by training the components individually against class labels. The resulting encoded representations would contain no information beyond class labels. With this, the saliency map [8, Eqs. (3) and (4), and Fig. 5] measures nothing more than the degree to which image regions impact classification accuracy of an object detector trained against class labels. Brain activity, whether encoded in EEG data or not, plays no role in constructing these saliency maps. The importance I_c of an EEG channel c as rendered in activation maps [8, Eqs. (5–7) and Fig. 6] measures nothing more than the degree to which removing the information in c decreases the classification accuracy, averaged over trials for a class and/or subjects. While this nominally is a valid approach, with the contaminated data collected with a block design, all these maps illustrate is the degree to which a given channel encodes the arbitrary long-term brain states associated with the block, not any class-specific information. Moreover, Tables 2, 3, and 8, suggest that any purported temporal information in Palazzo *et al.* [8, Figs. 7 and 9] is artifactual. Tables 5 and 7 suggest that activation maps computed with uncontaminated data collected with a rapid-event design would simply be blank, as accuracy would be at chance levels both prior and subsequent to removing the information in any particular EEG channel. Finally, association $A_{c,l}$ between an EEG channel c and any component l of an object detector is simply a linear combination of the class-average activation maps [8, Fig. 6] weighted by the degree to which removing the portion l of an object detector causes misclassifications to a given class. This holds whether l is a portion of a feature map, an entire feature map, or all feature maps in a given layer, as computed by Palazzo *et al.* [8, Eqs. (8–10)] and rendered in Palazzo *et al.* [8, Fig. 9]. The fact that the activation maps in Palazzo *et al.* [8, Fig. 9] become more diffuse for later layers in an object detector says nothing more than the fact that removing later layers in an object detector leads to higher entropy in the output distribution, a property solely due to the image classifier and completely independent of any brain processing, whether measured by EEG or not.

5.5 Summary

In summary, our results call into question not only the results of OP_1 but other published results as well [2], [3], [4], [5], [6], [7], [8]. They do so in four distinct ways. First, they raise doubts about all claims that depend directly or indirectly on the ability to use the kinds of classification algorithms reported here, including the particular classification algorithm of OP_1, to extract class information from the particular data of OP_1. That alone raises doubts about all of the above cited papers. Second, they raise doubts about the ability of the kinds of classification algorithms reported here, including the particular classification algorithm of OP_1, to extract class information from any EEG data collected with a block design. It places the burden of proof that there is no data contamination on any use of a block design. This raises doubts not just about the particular dataset collected by OP_1, but further about the experimental protocol proposed by OP_1. Third, they demonstrate that a whole spectrum of classification algorithms do not work on a dataset collected with a rapid-event design that does not suffer from data contamination. This raises doubts about not just the dataset and protocol, but further about the analysis methods and algorithms. Fourth, Sections 3.8 and 3.9 raise doubts about the general approach underlying the proposed methods and algorithms for using EEG data to advance computer vision. While we have employed the random-data attack to the particular methods of OP_1, we believe that it also can be applied to all of the methods in Palazzo *et al.* [3], Kavasidis *et al.* [4], Tirupattur *et al.* [7], and Palazzo *et al.* [8] as well. We are hindered in our attempt to conduct this analysis by the fact that the authors have declined to release their code to us, despite requests, and the fact that the published papers lack sufficient detail to replicate their models.

6 Related Work

To assess the degree to which the issues raised here impact other work in the field, we scrutinized the experiment designs

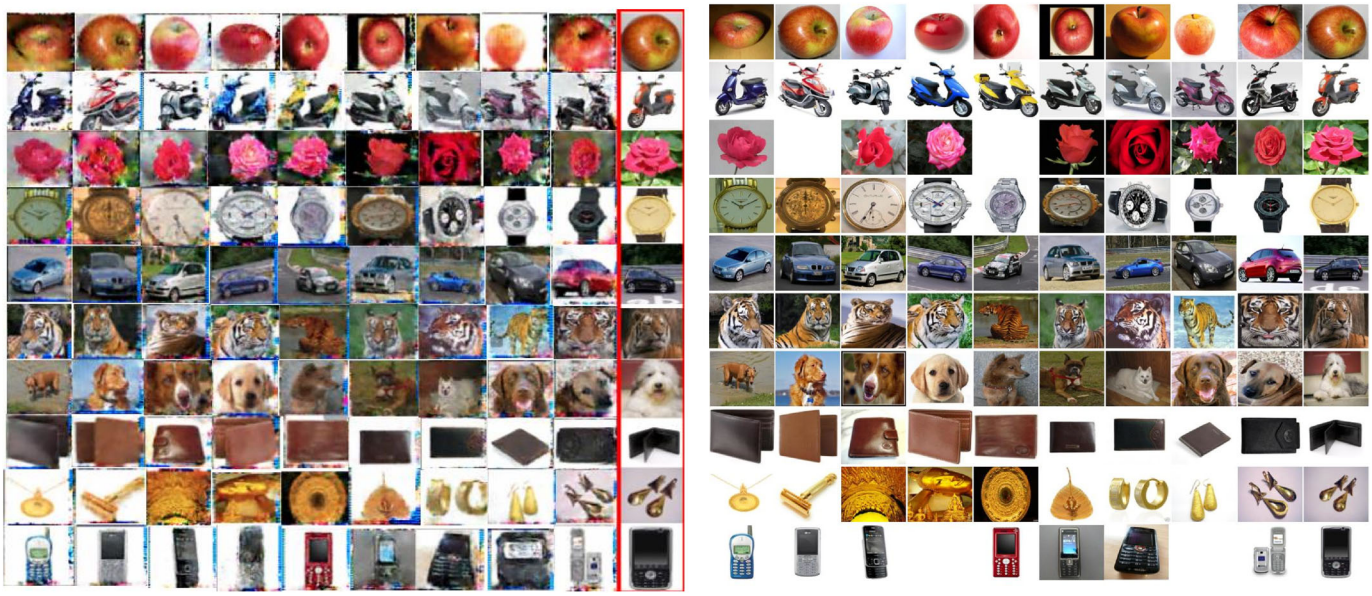

Fig. 2. (left) Fig. 6 from Tirupattur *et al.* [7] illustrating sample images purportedly generated by a GAN model from EEG encodings (except for the right column in red that illustrates a random image of the given class from the training data). (right) Corresponding identical ImageNet images for almost all of the generated images. Note that some, but not all, of the purportedly synthesized images on the left are horizontal mirrors of ImageNet images on the right. Also note that all of the purportedly synthesized images contain the same precise fine-grained detail as the corresponding ImageNet images. In particular, each image not only depicts the corresponding class but also depicts the exact non-class-specific background as the ImageNet counterpart.

in the 306 distinct papers that were either cited by a recent survey paper on deep-learning applied to EEG data [13] or that cited work critiqued here [1], [3], [4], [7] on Google Scholar at the time of writing. Of these, 180 appeared unrelated to the issue at hand as they do not collect or use EEG data for the purpose of classification. A further 4 were self citations already discussed. For the remaining 122, we attempted to determine the degree to which their results could be affected by the points we raise here. In particular, for each paper, we attempted to assess whether

- samples in the test set were recorded in close proximity to samples in the training set or
- the classes in the test trials appeared in the same order as classes in the training trials.

Unfortunately, many of the papers that we scrutinized lacked sufficient details of the experimental procedure to allow us to unequivocally answer the above two questions. About a third of the papers scrutinized appeared to be clear of the concerns raised here. About another third appeared to suffer from the concerns raised here. It was not possible to assess the remaining third. In the second category, we found ten papers that used the OP_1 data and are thus irrevocably flawed.

We further found a repeated and deeply concerning phenomenon. It is common today for authors to make neuroimaging datasets available to other researchers. We found about a dozen papers performed on such shared data, where the initial study that collected the data employed a block design that was carefully constructed to avoid data contamination. However, the subsequent study misused the data in ways that were not anticipated in the initial study that introduced data contamination.

We would be remiss if we did not mention additional design and method concerns that must be remedied in the field going forward. These include: using stimuli (images, video) that are not appropriately counterbalanced; using datasets to answer questions for which they were not designed; failure of those making datasets available to provide sufficient information for subsequent users to determine whether the dataset is appropriate for use with new applications; and failure to provide sufficient details of the procedure and method for transparency that would enable reproducibility of the study in accordance with the Open Science Framework [34].

7 Conclusion

The results in Tables 5 and 7 suggest that the ability to classify 40 object classes in image stimuli and 12 activity classes in video stimuli from an EEG signal is extremely difficult and well beyond the current state of the art. Moreover, the enterprise of using neuroimaging data to train better computer-vision systems, proposed by [35, Section 8 p. 625] and [29, Fig. 2 and Section 3 last paragraph p. 4068], requires more sophisticated methods than simply attaching a regressor to a pretrained object classifier and is also likely to be difficult and beyond the current state of the art. Both of these enterprises are the subject of substantial ongoing effort. When widely published [1], [2], [3], [4], [5], [6], [7], [8], inordinately optimistic claims can lead to misallocation of valuable resources and can sideline more modest but legitimate and important advances in the field. Thus, when the sensational claims are recognized as incorrect, it is imperative that the refutation be widely publicized to appropriately caution the community. Further, when the community as a whole appears to suffer from widespread but problematic practices, it is even more imperative that this warning be widely publicized to appropriately caution the community.

ACKNOWLEDGMENTS

This work was supported, in part, by the US National Science Foundation under Grants 1522954-IIS and 1734938-IIS, by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00341, and by Siemens Corporation, Corporate Technology. Any opinions, findings, views, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, official policies, or endorsements, either expressed or implied, of the sponsors. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

REFERENCES

[1] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, and M. Shah, "Deep learning human mind for automated visual classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6809–6817.

[2] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, M. Shah, and N. Souly, "Deep learning human mind for automated visual classification," 2016, *arXiv:1609.00344*.

[3] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, and M. Shah, "Generative adversarial networks conditioned by brain signals," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3410–3418.

[4] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, and M. Shah, "Brain2Image: Converting brain signals into images," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1809–1817.

[5] C. Du, C. Du, X. Xie, C. Zhang, and H. Wang, "Multi-view adversarially learned inference for cross-domain joint distribution matching," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1348–1357.

[6] P. Kumar, R. Saini, P. P. Roy, P. K. Sahu, and D. P. Dogra, "Envisioned speech recognition using EEG sensors," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 185–199, 2018.

[7] P. Tirupattur, Y. S. Rawat, C. Spampinato, and M. Shah, "ThoughtViz: Visualizing human thoughts using generative adversarial network," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 950–958.

[8] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, and M. Shah, "Decoding brain representations by multimodal learning of neural activity and visual features," 2018, *arXiv:1810.10974*.

[9] K. Linkenkaer-Hansen, V. V. Nikouline, J. M. Palva, and R. J. Ilmoniemi, "Long-range temporal correlations and scaling behavior in human brain oscillations," *J. Neurosci.*, vol. 21, no. 4, pp. 1370–1377, 2001.

[10] E. Bullmore *et al.*, "Colored noise and computational inference in neurophysiological (fMRI) time series analysis: Resampling methods in time and wavelet domains," *Hum. Brain Mapping*, vol. 12, pp. 61–78, 2001.

[11] A. M. Dale, "Optimal experimental design for event-related fMRI," *Hum. Brain Mapp.*, vol. 8, no. 2/3, pp. 109–114, 1999.

[12] E. Maris and R. Oostenveld, "Nonparametric statistical testing of EEG- and MEG-data," *J. Neurosci. Methods*, vol. 164, no. 1, pp. 177–190, 2007.

[13] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, 2019, Art. no. 051001.

[14] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, 2019, Art. no. 031001.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] B. Kaneshiro, M. P. Guimaraes, H.-S. Kim, A. M. Norcia, and P. Suppes, "A representational similarity analysis of the dynamics of object processing using single-trial EEG classification," *PLoS One*, vol. 10, no. 8, 2015, Art. no. e0135697.

[18] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2929–2936.

[19] H. Cecotti and A. Gräser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 433–445, Mar. 2011.

[20] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from EEG with deep recurrent-convolutional neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

[21] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, "Deep feature learning for EEG recordings," 2015, *arXiv:1511.04306*.

[22] C. Wang, S. Xiong, X. Hu, L. Yao, and J. Zhang, "Combining features from ERP components in single-trial EEG for discriminating four-category visual objects," *J. Neural Eng.*, vol. 9, no. 5, 2012, Art. no. 056013.

[23] T. A. Carlson, H. Hogendoorn, R. Kanai, J. Mesik, and J. Turret, "High temporal resolution decoding of object position and category," *J. Vis.*, vol. 11, no. 10, 2011, Art. no. 9.

[24] I. Simanova, M. Van Gerven, R. Oostenveld, and P. Hagoort, "Identifying object categories from event-related EEG: Toward decoding of conceptual representations," *PLoS One*, vol. 5, no. 12, 2010, Art. no. e14465.

[25] N. Bigdely-Shamlo, A. Vankov, R. R. Ramirez, and S. Makeig, "Brain activity-based image classification from rapid serial visual presentation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 5, pp. 432–441, Oct. 2008.

[26] A. X. Stewart, A. Nuthmann, and G. Sanguinetti, "Single-trial classification of EEG in a visual object task using ICA and machine learning," *J. Neurosci. Methods*, vol. 228, pp. 1–14, 2014.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[28] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," *Uncertainty Artif. Intell.*, pp. 266–273, 2011.

[29] J. M. Siskind, "Conducting neuroscience to guide the development of AI," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 4067–4072.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[33] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[34] E. D. Foster and A. Deardorff, "Open science framework (OSF)," *J. Med. Library Assoc.*, vol. 105, no. 2, pp. 203–206, 2017.

[35] A. Barbu *et al.*, "Seeing is worse than believing: Reading people's minds better than computer-vision methods recognize actions," in *Proc. Eur. Conf. Comput. Vis.*, 2014, vol. 5, pp. 612–627.

[36] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

Ren Li received the BS degree in electrical engineering from the University of Science and Technology of China, Anhui, China, in 2016. He is currently working toward the PhD degree in electrical and computer engineering at Purdue University, West Lafayette, Indiana. His research interests include computer vision and machine learning, especially benefiting computer-vision systems from brain-derived information.

Jared S. Johansen received the BS degree in electrical engineering from Brigham Young University, Provo, Utah, in 2010, the MS degree in electrical engineering and the MBA degree from the University of Utah, Salt Lake City, Utah, in 2012. He is currently working toward the PhD degree in electrical and computer engineering at Purdue University, West Lafayette, Indiana. His research lies at the intersection of machine learning, computer vision, natural language processing, and robotics.

Hamad Ahmed received the BS degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan. He is currently working toward the PhD degree in electrical and computer engineering at Purdue University, West Lafayette, Indiana. His research interests lie at the intersection of machine learning and computer vision.

Thomas V. Ilyevsky received the BS degree in electrical and computer engineering from Cornell University, Ithaca, New York, in 2016. He is currently working toward the PhD degree in electrical and computer engineering at Purdue University, West Lafayette, Indiana. His research focuses on artificial intelligence, computer vision, and natural language processing in the context of human-computer interaction.

Ronnie B. Wilbur received the BA degree from the University of Rochester, Rochester, New York and the PhD degree from the University of Illinois at Urbana-Champaign, Champaign, Illinois, in linguistics. She is a professor at Purdue University, West Lafayette, Indiana, in the Department of Speech, Language, and Hearing Sciences and the Department of Linguistics. Prior to coming to Purdue, she was visiting faculty at the University of Southern California, Los Angeles, California and faculty at Boston University, Boston, Massachusetts. She has been widely invited as visiting professor (Amsterdam, Graz, Zagreb, Paris ENS, Salzburg, Stuttgart) and has been funded by both the National Science Foundation and the National Institutes of Health. Her theoretical and experimental research on linguistics of sign languages currently seeks to apply computer vision/AI, neuroimaging, and neurophysiology toward automatic recognition of sign languages.

Hari M. Bharadwaj received the BTech degree in electrical engineering from the Indian Institute of Technology, Madras, India, in 2006, the MS degrees in electrical engineering and biomedical engineering from the University of Michigan, Ann Arbor, Michigan, in 2008, and the PhD degree in biomedical engineering from Boston University, Boston, Massachusetts, in 2014. He is an assistant professor at Purdue University, West Lafayette, Indiana, in the Department of Speech, Language, and Hearing Sciences, and the Weldon School of Biomedical Engineering. Following post-doctoral work at the Martinos Center for Biomedical Imaging at Massachusetts General Hospital, Boston, Massachusetts, he joined the faculty at Purdue University, West Lafayette, Indiana in 2016, where his lab integrates a multidisciplinary array of tools to investigate the neural mechanisms underlying auditory perception in humans.

Jeffrey Mark Siskind (Senior Member, IEEE) received the BA degree in computer science from the Technion, Israel Institute of Technology, Haifa, Israel, in 1979, the SM degree in computer science from MIT, Cambridge, Massachusetts, in 1989, and the PhD degree in computer science from MIT, Cambridge, Massachusetts, in 1992. He did a post-doctoral fellowship at the University of Pennsylvania Institute for Research in Cognitive Science, Philadephia, Pennsylvania, from 1992 to 1993. He was an assistant professor at the University of Toronto Department of Computer Science, Toronto, Canada, from 1993 to 1995, a senior lecturer at the Technion Department of Electrical Engineering, Haifa, Israel, in 1996, a visiting assistant professor at the University of Vermont Department of Computer Science and Electrical Engineering, Burlington, Vermont, from 1996 to 1997, and a research scientist at NEC Research Institute, Inc., Princeton, New Jersey, from 1997 to 2001. He joined the Purdue University School of Electrical and Computer Engineering, West Lafayette, Indiana, in 2002 where he is currently a professor.