Structured Neural Decoding With Multitask Transfer Learning of Deep Neural Network Representations

Changde Du, Changying Du, Lijie Huang, Haibao Wang, and Huiguang He, *Senior Member, IEEE*

***Abstract*—The reconstruction of visual information from human brain activity is a very important research topic in brain decoding. Existing methods ignore the structural information underlying the brain activities and the visual features, which severely limits their performance and interpretability. Here, we propose a hierarchically structured neural decoding framework by using multitask transfer learning of deep neural network (DNN) representations and a matrix-variate Gaussian prior. Our framework consists of two stages, Voxel2Unit and Unit2Pixel. In Voxel2Unit, we decode the functional magnetic resonance imaging (fMRI) data to the intermediate features of a pretrained convolutional neural network (CNN). In Unit2Pixel, we further invert the predicted CNN features back to the visual images. Matrix-variate Gaussian prior allows us to take into account the structures between feature dimensions and between regression tasks, which are useful for improving decoding effectiveness and interpretability. This is in contrast with the existing single-output regression models that usually ignore these structures. We conduct extensive experiments on two real-world fMRI data sets, and the results show that our method can predict CNN features more accurately and reconstruct the perceived natural images and faces with higher quality.**

***Index Terms*—Deep neural network (DNN), functional magnetic resonance imaging (fMRI), image reconstruction, multioutput regression, neural decoding.**

Manuscript received July 6, 2019; revised November 28, 2019, March 23, 2020, and September 21, 2020; accepted September 24, 2020. Date of publication October 19, 2020; date of current version February 4, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61976209, Grant 62020106015, Grant 61906188, and Grant 61602449; in part by the Chinese Academy of Sciences (CAS) International Collaboration Key Project under Grant 173211KYSB20190024; and in part by the Strategic Priority Research Program of CAS under Grant XDB32040000. *(Corresponding author: Huiguang He.)*

Changde Du is with the Research Center for Brain-Inspired Intelligence, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Huawei Cloud BU EI Innovation Laboratory, Beijing 100085, China (e-mail: duchangde@gmail.com).

Changying Du is with the Huawei Noah's Ark Laboratory, Beijing 100085, China (e-mail: ducyatict@gmail.com).

Lijie Huang is with the Research Center for Brain-Inspired Intelligence, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lijie.huang@ia.ac.cn).

Haibao Wang is with the Research Center for Brain-Inspired Intelligence, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: haibaow@hotmail.com).

Huiguang He is with the Research Center for Brain-Inspired Intelligence, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: huiguang.he@ia.ac.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3028167

I. Introduction

THE advance in sensory neuroscience could lead to new insights into brain function and aid efforts to improve the artificial intelligence [1]. One of the critical aspects to the research is neural decoding, which aims to build the relationship between visual contents and the corresponding functional magnetic resonance imaging (fMRI) brain recordings. The most accurate visual stimulus reconstruction methods in existence rely on convolutional neural network (CNN) [2], [3], which are pretrained on the large-scale image data set for visual recognition tasks. In recent years, many studies have used the intermediate features of pretrained CNN as the medium for neural decoding. For example, researchers have tried to reconstruct human faces [4], handwritten characters [5], or natural images [3] using the CNN features predicted from the multivariate fMRI data.

In the neural decoding task of using pretrained CNN, the most important point is how to model the mapping relationship between fMRI voxels and CNN features. Existing neural decoding methods assume that different intermediate features of CNN are independent of each other, and they establish a separate prediction model for each CNN feature to decode the fMRI voxels [2], [6]. However, multiple intermediate features of CNN are not independent of each other, and they are usually correlated with each other through some potential structures. In addition, there is also some correlation between the voxel features of fMRI data, which can reflect the visual stimulus information perceived by subjects to a certain extent. Finally, taking the learning of the single-output prediction model of each CNN feature as a task, there are also some dependencies among multiple tasks. Constructing multitask learning method (by applying appropriate constraints to the regression weights) is also helpful to improve the accuracy of fMRI decoding. We refer to the aforementioned dependencies between data dimensions and between learning tasks as structured information. Because of ignoring these structural information, the performance and interpretability of traditional neural decoding methods are limited. Therefore, we advocate that these structured information should be fully utilized in the neural decoding process using CNN intermediate features.

On the other hand, it is also a very important problem how to accurately reconstruct the corresponding visual images by using the intermediate features of CNN obtained by fMRI data decoding. In the literature, this problem is solved by the maximum *a posterior* (MAP) estimation strategy. For example, gradient-based optimization methods can be applied to find an optimal solution in the image space, so that the CNN features corresponding to the optimal image are as close as possible to the target CNN features. In the image optimization process, some constraints can be imposed to speed up the convergence process, such as total variation (TV) regularizer [7] or pretrained deep image generative model [3], [8]. In another way, a deconvolutional neural network (De-CNN) can also be trained based on a large number of image-CNN feature data pairs to predict the corresponding image according to the input CNN feature [9]. Unfortunately, both of these solutions have their drawbacks. The inference process in the first solution is very slow, while the second solution tends to produce blurry, low-quality images, which we do not want to see. Recently, it is an exciting way to use the conditional deep generative models (DGMs) such as conditional variational autoencoders (CVAEs) [10] and conditional generative adversarial nets (CGANs) [11], [12] to invert the CNN features back to visual images. In particular, thanks to the great success of adversarial learning in high-fidelity image synthesis [13], [14], we believe that conditional adversarial image synthesis method will also greatly improve the effect of visual image reconstruction.

In this article, we propose a novel structured neural decoding framework for the perceived image reconstruction. The key idea is to use features derived from a CNN trained on plentiful image data to improve the reconstruction of images from fMRI data (where data are scarce). This process can transfer information from deep neural network (DNN) representations to brain activity data with the goal of image reconstruction. Specifically, our framework consists of two stages, Voxel2Unit and Unit2Pixel (see Fig. 1). In Voxel2Unit, we first use the structured multioutput regression (SMR) model to decode the fMRI voxel features to the intermediate CNN features. In Unit2Pixel, we further use the introspective conditional generation (ICG) model to invert the predicted CNN features back to the visual images. Our main contributions include

1) We use matrix-variable Gaussian prior to establish an SMR model to decode the multivariate fMRI data to the CNN features.
2) We use variational adversarial learning to build an ICG model to reconstruct high-quality images based on the decoded CNN features.
3) For validating the performance of the proposed framework, we collected a new fMRI data set[1] evoked by 800 different face stimuli.
4) We studied the relationship between different feature layers of three CNN architectures and different brain visual areas and found that there is a homology between computer and human vision.
5) The experiments demonstrate that the proposed approach can accurately reconstruct the perceived natural images and human faces from brain activity.

II. Related Work

A. DNN-Based Neural Decoding

Neural decoding studies can be divided into three categories, i.e., *semantic classification* [15]–[19], *image identification* [6], [20], and *image reconstruction* [21]–[23]. Although a lot of DNN-based neural decoding methods have been proposed for image reconstruction in recent years [2]–[5], their performances still need to be improved. For example, Wen *et al.* [2] first used linear regression to transform fMRI data into higher-level semantic features of CNN, and then trained a separate image decoding network to reconstruct natural images. Du *et al.* [5] proposed a multiview deep generative model (DGMM), in which they first used the sparse linear model to map the brain activities to the latent representation of the variational autoencoder (VAE), and then used the VAE decoder to reconstruct the image. But it is difficult to use DGMM to reconstruct the natural images clearly due to the natural shortcomings of VAE. More recently, Shen *et al.* [3] also used linear regression to decode the fMRI data into the semantic feature vector of CNN, and then used the gradient descent method to iteratively find the optimal image for each test sample.

Most of the above methods are based on a common assumption that the internal units of CNN are independent of each other. Therefore, they need to fit many independent single-output linear regression (SLR) models, each of which is used to predict the feature of a CNN unit. Unlike them, our two-stage decoding framework first adopts an SMR model to decode the multiple CNN features from fMRI data, and then uses an ICG model to reconstruct the visual image from the decoded CNN features.

B. Multioutput Regression

Our SMR model is a more general multioutput regression framework, and many existing models [24]–[27] can be regarded as the special cases of it. Rothman *et al.* [24] proposed a multivariate regression with covariance estimation (MRCE) model, in which only the correlation between outputs is considered. In [25], the MRCE model is extended by simultaneously exploiting the correlation between outputs and tasks, but the correlation between inputs is ignored. Furthermore, many previous multitask learning lie in discovering the relationship among the tasks by mining the common input structures shared by the tasks [28]–[30]. The regularization method is widely used to discover the relationship among the tasks [31]–[33]. Argyriou *et al.* [31] penalized the regression weights by using the spectral functions to learn the feature structure shared across the tasks. Chen *et al.* [33] penalized the regression weights by combining the nuclear norm and the $\ell_{p,q}$ norm to simultaneously learn the sparsity of the regression weights and the correlation among the tasks. The structural constraints between different regression tasks and between different output dimensions are considered at the same time via the inverse-covariance regularization in [26]. The above multitask learning approaches assume that the relevant inputs corresponding to each output are identical. This is unreasonable in neural decoding applications because previous study has shown that there is a corresponding relationship between the hierarchical visual representation of images and the neural expression in various stages of human visual processing [34]–[36].

[1] available at https://figshare.com/articles/dataset/FaceBold/13019966

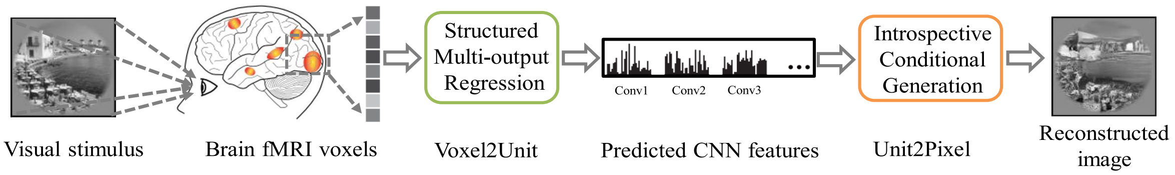

Fig. 1. Overview of the proposed hierarchically structured neural decoding framework. It involves two cascaded stages, 1) *Voxel2Unit*: decoding the CNN features from fMRI activity and 2) *Unit2Pixel*: reconstructing the perceived image using the decoded CNN features.

C. Deep Generative Models

VAEs [37] and generative adversarial nets (GANs) [38] are the most popular DGMs, and they have shown great success in generating high-quality images [13], [14], [39]. Recently, some research studies have applied VAEs and GANs to neural decoding [4], [40]–[42]. For example, Du *et al.* [40] proposed a VAE-based image reconstruction framework, and Güçlütürk *et al.* [4] proposed a GAN-based framework. Both VAE-based and GAN-based neural decoding methods have their own strengths and limitations in image reconstruction. VAE-based neural decoding methods are theoretically elegant and easy to train but the reconstructed images are often blurry. GAN-based neural decoding methods usually produce much clearer reconstructions but face challenges in training stability. Our image reconstruction model can be seen as a hybrid of VAE and GAN, which combines the best of both worlds.

III. Hierarchically Structured Neural Decoding

Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N\times M}$ represent the visual images, $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]^\top \in \mathbb{R}^{N\times K}$ represent the intermediate CNN features (we use the pretrained AlexNet [43] model) of $\mathbf{Y}$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N\times D}$ represent the corresponding multivariate fMRI data. Here, N is the number of training data, M, K, and D are the corresponding data dimensions, respectively. The commonly used symbols and their definitions are listed in Table I. The correlation between distinct CNN features is called ***output structure***, the correlation between distinct fMRI voxel features is called ***input structure***, and the correlation between distinct single-output regression tasks is called ***task structure***. The proposed hierarchically structured neural decoding framework is illustrated in Fig. 1.

A. Voxel2Unit: SMR

In order to improve the accuracy and interpretability of neural decoding, a structured multiple output regression (SMR) model was established in this stage to simultaneously model the three structural information mentioned above. As shown in Fig. 2, the learning goal of multioutput regression model is to establish the linear mapping relationship between multivariable input $\mathbf{x}_n \in \mathbb{R}^D$ and multivariable output $\mathbf{h}_n \in \mathbb{R}^K$ via

$$\mathbf{h}_n = \mathbf{W}^\top \mathbf{x}_n + \mathbf{b} + \boldsymbol{\epsilon}_n \quad \forall n = 1, \ldots, N. \tag{1}$$

Here, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{D\times K}$ is the regression coefficient matrix where its column $\mathbf{w}_k \in \mathbb{R}^D$ is the coefficient of the kth output and its row $\mathbf{w}^d \in \mathbb{R}^K$ is the corresponding coefficient of the dth input. $\mathbf{b} = [b_1, \ldots, b_K]^\top \in \mathbb{R}^K$ is the bias vector for the K outputs, and $\boldsymbol{\epsilon}_n = [\epsilon_{n1}, \ldots, \epsilon_{nK}]^\top \in \mathbb{R}^K$ is the Gaussian noise vector.

TABLE I

Definition of Frequently Used Symbols

Symbol	Definition
N	Number of training instances
$\mathbf{X}$	Matrix of fMRI activity patterns
$\mathbf{Y}$	Matrix of visual images
$\mathbf{H}$	Matrix of intermediate CNN features
$\mathbf{x}_n$	n-th fMRI activity pattern
$\mathbf{h}_n$	n-th intermediate CNN features
M	Dimension of visual image
D	Dimension of fMRI activity pattern
K	Dimension of intermediate CNN features
$\mathcal{N}(\cdot)$	Gaussian distribution
$\mathcal{MN}(\cdot)$	Matrix-variate Gaussian distribution
$\mathbf{W}$	Matrix of regression weights $\mathbf{W} \in \mathbb{R}^{D\times K}$
$\mathbf{b}$	Vector of bias $\mathbf{b} \in \mathbb{R}^K$
$\boldsymbol{\Omega}$	Full covariance matrix $\boldsymbol{\Omega} \in \mathbb{R}^{K\times K}$
$\boldsymbol{\Sigma}_r$	Row covariance matrix $\boldsymbol{\Sigma}_r \in \mathbb{R}^{D\times D}$
$\boldsymbol{\Sigma}_c$	Column covariance matrix $\boldsymbol{\Sigma}_c \in \mathbb{R}^{K\times K}$
$\lambda, \lambda_1, \lambda_2, \lambda_3$	Hyperparameters of SMR model
α, β	Hyperparameters of ICG model
$\Vert\cdot\Vert_1$	ℓ_1-norm operator
$\vert\cdot\vert$	Matrix determinant operator
$\mathrm{tr}(\cdot)$	Matrix trace operator
$\otimes$	Kronecker product operator
$\mathrm{vec}(\cdot)$	Vectorization of a matrix
$\mathbf{y}$	Real visual image
$\mathbf{y}_r$	Reconstructed visual image
$\mathbf{y}_g$	Generated visual image
$\mathbf{y}_f$	Fake visual image
$\mathbf{z}$	Latent variable in VAEs or GANs
θ	Parameters of decoder/generator
ϕ	Parameters of encoder

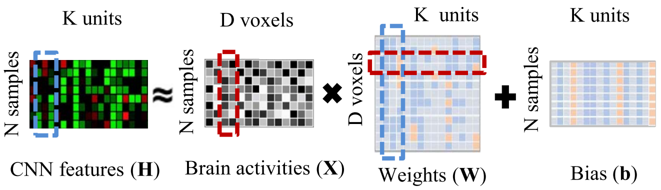

Fig. 2. Voxel2Unit: SMR. The red and blue dashed rectangles represent the possible dependencies between the inputs and the outputs, respectively.

1) Prior With Input and Task Structures: Considering the dependence between fMRI voxels and the dependence between single-output regression tasks, we apply the following structural prior to $\mathbf{W}$:

$$p(\mathbf{W}) = \left(\prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_k|\mathbf{0}, \mathbf{I}_D)\right) \mathcal{MN}(W|\mathbf{0}_{D\times K}, \boldsymbol{\Sigma}_r, \boldsymbol{\Sigma}_c) \tag{2}$$

where $\mathbf{I}_D$ is a $D \times D$ identity matrix i.e., its diagonal elements are one and all the other elements are zero, $\mathbf{0} \in \mathbb{R}^D$ is a D-dimensional vector with zero entries, $\mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$ denotes a matrix-variate Gaussian distribution [44] with mean $\mathbf{M} \in \mathbb{R}^{D\times K}$, row covariance matrix $\mathbf{A} \in \mathbb{R}^{D\times D}$ and column covariance matrix $\mathbf{B} \in \mathbb{R}^{K\times K}$. In (2), $\mathcal{N}(\mathbf{w}_k|\mathbf{0}, \mathbf{I}_D)$ regularizes the weight vector $\mathbf{w}_k$ *individually*, while $\mathcal{MN}(W|\mathbf{0}_{D\times K}, \mathbf{\Sigma}_r, \mathbf{\Sigma}_c)$ *couples* the D rows of $\mathbf{W}$ by the covariance matrix $\mathbf{\Sigma}_r$, and the K columns of $\mathbf{W}$ by the covariance matrix $\mathbf{\Sigma}_c$. As a result, we can model the *input structure* and *task structure* by learning $\mathbf{\Sigma}_r$and $\mathbf{\Sigma}_c$, respectively.

2) Likelihood With Output Structure: However, due to the limited expression capacity of the linear regression function, the multioutput model based on the above matrix-variable Gaussian prior may not be able to fully characterize the correlation between K outputs. In order to completely characterize the potentially remaining structural information among the K outputs that is not explained by the task structure, we impose a full covariance matrix $\mathbf{\Omega} \in \mathbb{R}^{K\times K}$ on the output Gaussian noise distribution. For N training instances, the likelihood function can be written as

$$p(\mathbf{H}|\mathbf{X}, \mathbf{W}, \mathbf{b}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{h}_n|\mathbf{W}^\top \mathbf{x}_n + \mathbf{b}, \mathbf{\Omega}). \tag{3}$$

Note that most previous neural decoding methods [2], [3], [6] assume the output Gaussian noise distribution has a diagonal covariance (i.e., $\mathbf{\Omega} = \mathbf{I}$). Therefore, they cannot effectively utilize the output structure information in the K outputs.

Given the structural prior in (2) and the likelihood function in (3), the posterior distribution of $\mathbf{W}$ can be written as

$$\begin{aligned} &p(\mathbf{W}|\mathbf{X}, \mathbf{H}, \mathbf{b}, \mathbf{\Omega}, \mathbf{\Sigma}_r, \mathbf{\Sigma}_c) \\ &\quad\propto \left(\prod_{n=1}^{N} \mathcal{N}\big(\mathbf{h}_n|\mathbf{W}^\top \mathbf{x}_n + \mathbf{b}, \mathbf{\Omega}\big)\right) \\ &\quad\cdot \left(\prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_k|\mathbf{0}, \mathbf{I}_D)\right) \mathcal{MN}(W|\mathbf{0}_{D\times K}, \mathbf{\Sigma}_r, \mathbf{\Sigma}_c). \end{aligned} \tag{4}$$

It is intractable to use Bayesian estimation for the above posterior distribution, and we apply the point estimation to the regression coefficient matrix $\mathbf{W}$. Taking the log of (4) and ignoring the constants, we can solve $\mathbf{W}$ by MAP estimation. Specifically, the negative log-posterior of $\mathbf{W}$ can be written as

$$\begin{aligned} \mathcal{J} &= \mathrm{tr}((\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)\mathbf{\Omega}^{-1}(\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)^\top) \\ &\quad - N \log |\mathbf{\Omega}^{-1}| + \lambda \mathrm{tr}(\mathbf{WW}^\top) + \lambda_1 \mathrm{tr}\big(\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{\Sigma}_c^{-1}\mathbf{W}^\top\big) \\ &\quad - K \log \big|\mathbf{\Sigma}_r^{-1}\big| - D \log \big|\mathbf{\Sigma}_c^{-1}\big| \end{aligned} \tag{5}$$

where $\mathrm{tr}(\cdot)$ denotes matrix trace, $\mathbf{1}$ denotes a $N \times 1$ vector of all 1 s, and $|\cdot|$ denotes matrix determinant. Here, λ and λ_1 are the regularization hyperparameters. Note that the $\mathrm{tr}(\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{\Sigma}_c^{-1}\mathbf{W}^\top)$ term captures the dependencies among the rows of $\mathbf{W}$ by learning the feature inverse covariance matrix $\mathbf{\Sigma}_r^{-1}$, and the dependencies among the columns of $\mathbf{W}$ by learning the task inverse covariance matrix $\mathbf{\Sigma}_c^{-1}$.

3) Sparse Covariance Selection: The inverse covariance matrices $\mathbf{\Omega}^{-1}$, $\mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$ are expected to be sparse for two reasons: 1) sparsity leads to improved robust estimates of them [45] and 2) sparsity supports the assumption that dependencies between features/outputs/tasks tend to be sparse, i.e., not all pairs of voxels/units/tasks are related. For example, when $\mathbf{\Sigma}_r^{-1}$ is sparse, a zero entry in it indicates no direct interaction between the two corresponding voxels in the multioutput regression. For sparse $\mathbf{\Sigma}_c^{-1}$ and $\mathbf{\Omega}^{-1}$, we have similar explanations. Therefore, we impose sparsity constraints on $\mathbf{\Omega}^{-1}$, $\mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$ via the ℓ_1 penalty. Let $\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{\Omega}^{-1}, \mathbf{\Sigma}_r^{-1}, \mathbf{\Sigma}_c^{-1}\}$, the ℓ_1 regularized objective function can be written as

$$\min_{\Theta} \mathcal{J}_s = \mathcal{J} + \lambda_2 ||\mathbf{\Omega}^{-1}||_1 + \lambda_3\big(\big\|\mathbf{\Sigma}_r^{-1}\big\|_1 + \big\|\mathbf{\Sigma}_c^{-1}\big\|_1\big) \tag{6}$$

where $||\cdot||_1$ denotes the ℓ_1-norm, and $\{\lambda_2, \lambda_3\}$ are the hyperparameters, which control the sparsity of the inverse covariance matrices.

B. Unit2Pixel: ICG

In this stage, our goal is to reconstruct the preferred image for the CNN features decoded from the first stage. Typically, this problem can be regarded as an optimization problem, that is, using gradient descent method to find an optimal image, so that its CNN features are the closest to the target [3], [7]. However, this approach is very slow because it needs to be optimized independently for each sample.

1) Conditional DGMs: Recently, it is an exciting method to directly generate the corresponding visual images by using conditional DGMs such as CVAEs [10] and CGANs [11], [12] under the condition of given CNN features. For example, treating the predicted CNN features $\mathbf{H}$ as condition, the objective function of CVAE can be written as

$$\mathcal{L}_{\mathrm{CVAE}} = \underbrace{-\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})}[\log p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})]}_{L_{\mathrm{AE}}} + \underbrace{\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})||p(\mathbf{z}))}_{D_{\mathrm{KL}}} \tag{7}$$

where $p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})$ denotes the generative network with parameter θ, $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})$ denotes the inference network with parameter ϕ, $\mathbf{y}$ is the visual images, $\mathbf{z}$ is the latent variables, and $\mathbf{h}$ is the given condition. The overview of CVAE is shown in the gray sub-panel of Fig. 3. In (7), the first term L_{AE} denotes the image reconstruction error, and D_{KL} is a regularization term that narrows the distance between the approximate posterior $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})$, and the prior $p(\mathbf{z})$. However, both CVAEs and CGANs have their own significant strengths and limitations in image generation. CVAE has stable training process, but its generated images are relatively blurry. In contrast, CGANs can produce relatively high-quality images, but its training process is prone to lose stability

2) Introspective Adversarial Learning: To address these problems, we build an ICG model, and the illustration is shown in Fig. 3. Recall that, in CVAE, D_{KL} is minimized along with L_{AE} on the observable data points (see 7). Nevertheless, in introspective adversarial training, D_{KL} is adversarially optimized along with L_{AE}. Specifically, for real images, we still

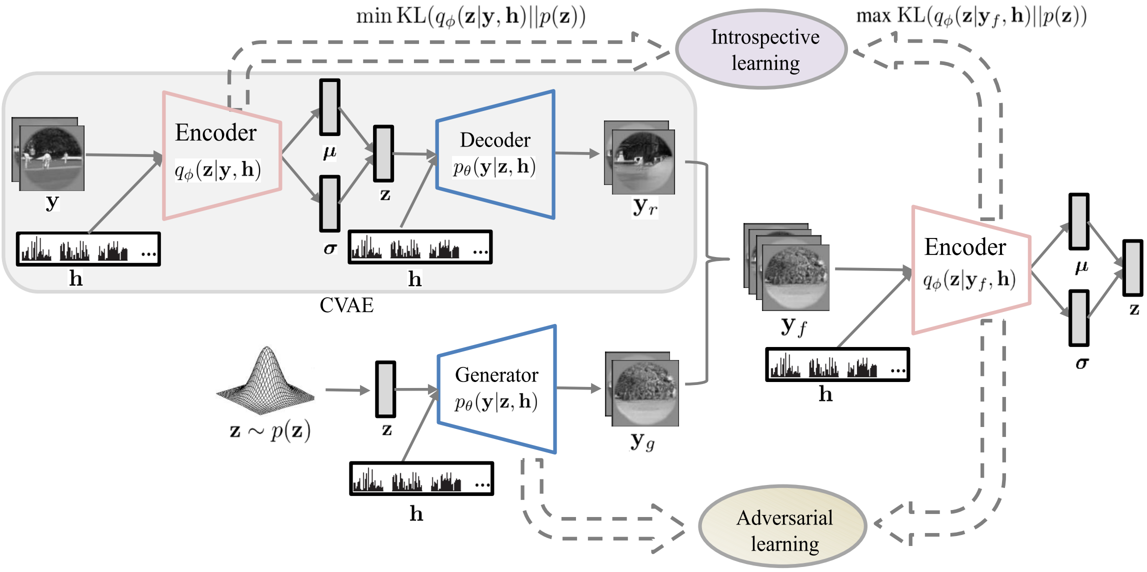
min KL($q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})||p(\mathbf{z})$)
Introspective learning
max KL($q_\phi(\mathbf{z}|\mathbf{y}_f,\mathbf{h})||p(\mathbf{z})$)
Encoder $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})$
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
$\mathbf{z}$
Decoder $p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})$
$\mathbf{y}$
$\mathbf{y}_r$
$\mathbf{h}$
CVAE
Encoder $q_\phi(\mathbf{z}|\mathbf{y}_f,\mathbf{h})$
$\mathbf{y}_f$
Generator $p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})$
$\mathbf{z} \sim p(\mathbf{z})$
$\mathbf{y}_g$
Adversarial learning

Fig. 3. Unit2Pixel: ICG. In the training phase, $\mathbf{y}$ comes from large-scale image data (including images without fMRI), and $\mathbf{h}$ is the correspondingly true CNN features. Network parameters are shared between the decoder and the generator, and similarly for the two encoders. In the test phase, we use the generator $p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})$ to obtain the image reconstructions, where $\mathbf{z} \sim p(\mathbf{z})$ and $\mathbf{h}$ is the decoded CNN features.

minimize D_{KL}, but for the reconstructed or generated (we call them fake images) ones, we maximize D_{KL}. The encoder and decoder play an adversarial learning game, where the encoder attempts to minimize D_{KL} for real images while maximize it for the fakes, while the decoder attempts to mislead the encoder by minimizing D_{KL} for the fakes. The parameters of the decoder and encoder can be iteratively optimized by the following two formulas:

$$\hat{\theta} = \arg\min_{\theta}[L_{\mathrm{AE}} + \alpha D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{y}_f,\mathbf{h})||p(\mathbf{z}))] \tag{8}$$

$$\begin{aligned}\hat{\phi} = \arg\min_{\phi}[&L_{\mathrm{AE}} + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})||p(\mathbf{z}))\\ &- \alpha D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{y}_f,\mathbf{h})||p(\mathbf{z}))]\end{aligned} \tag{9}$$

where $\mathbf{y}_f$ denotes the fake images sampled from $p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h})$. Equation (8) and (9) form a specific hybrid model of CVAE and CGAN. α and β are hyperparameters, balancing the impact of CVAE and CGAN. Our method will degrade to standard CVAE when $\alpha = 0, \beta = 1$, and to a regularized CGAN when $\alpha = 1, \beta = 0$.

Cooperative learning and adversarial learning coexist in the procedure of optimizing (8) and (9). For the real images, (8) and (9) cooperatively minimize L_{AE} and $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})||p(\mathbf{z}))$, which is equivalent to the optimization of CVAE. For the fake images, (8) and (9) form the CGAN-like adversarial learning game, in which (8) attempts to minimize $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{y}_f,\mathbf{h})||p(\mathbf{z}))$ while (9) attempts to maximize it. Here, $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{h})$ plays two different roles simultaneously. On the one hand, it acts as the encoder of CVAE, on the other hand, it acts as the discriminator of CGAN. Compared with other VAE and GAN hybrid models [46], [47], our method does not require additional discriminators (since the encoder also played the role of discriminator), which reduced the number of model parameters.

IV. Optimization

End-to-end training the proposed two-stage framework requires a large amount of paired image-fMRI data, which cannot be satisfied in most cases. Instead, we train each stage individually, which has two advantages: 1) after the first stage of training, we can select some CNN units with high decodability for the second stage of training and 2) we can use a large amount of additional image data in the second stage to augment the training data set.

A. Training SMR Model

We adopt an alternating optimization strategy to learn the proposed SMR model. In each iteration, we alternatively optimize one variable with others fixed.

1) Update $\mathbf{W}$*:* We solve the following subproblem to update $\mathbf{W}$ with $\mathbf{b}$, $\boldsymbol{\Omega}^{-1}$, $\boldsymbol{\Sigma}_r^{-1}$ and $\boldsymbol{\Sigma}_c^{-1}$ fixed:

$$\begin{aligned}\min_{\mathbf{W}}\ \mathcal{L}_{\mathbf{W}} = \mathrm{tr}((&\mathbf{H}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1}(\mathbf{H}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^\top)^\top)\\ &+\lambda\mathrm{tr}(\mathbf{W}\mathbf{W}^\top)+\lambda_1\mathrm{tr}\big(\boldsymbol{\Sigma}_r^{-1}\mathbf{W}\boldsymbol{\Sigma}_c^{-1}\mathbf{W}^\top\big).\end{aligned} \tag{10}$$

The following proposition characterizes its optimal solution.

Proposition 1: The optimal solution of (10) satisfies $\mathrm{vec}(\hat{\mathbf{W}}) = \mathbf{U}^{-1}\mathbf{V}$, where $\mathbf{U} = \boldsymbol{\Omega}^{-1}\otimes(\mathbf{X}^\top\mathbf{X})+\lambda\mathbf{I}_{KD}+\lambda_1\boldsymbol{\Sigma}_c^{-1}\otimes\boldsymbol{\Sigma}_r^{-1}$ and $\mathbf{V} = \mathrm{vec}(\mathbf{X}^\top(\mathbf{H}-\mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1})$.

See Appendix A for detailed proof. Here, $\otimes$ denotes the Kronecker product and $\mathrm{vec}(\mathbf{W})$ denotes the vectorization of $\mathbf{W}$. When faced with relative large K and D, we use gradient descent method with $\nabla_{\mathbf{W}}\mathcal{L}_{\mathbf{W}} = \mathbf{X}^\top(\mathbf{H}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1} + \lambda\mathbf{W}+\lambda_1\boldsymbol{\Sigma}_r^{-1}\mathbf{W}\boldsymbol{\Sigma}_c^{-1}$ to solve it.

2) Update $\mathbf{b}$*:* Given $\mathbf{W}$, $\boldsymbol{\Omega}^{-1}$, $\boldsymbol{\Sigma}_r^{-1}$ and $\boldsymbol{\Sigma}_c^{-1}$, the bias vector $\mathbf{b}$ can be obtained by

$$\min_{\mathbf{b}}\ \mathrm{tr}((\mathbf{H}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1}(\mathbf{H}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^\top)^\top). \tag{11}$$

The result is $\hat{\mathbf{b}} = (1/N)(\mathbf{H}-\mathbf{X}\mathbf{W})^\top\mathbf{1}$.

Algorithm 1 SMR Training (Voxel2Unit Stage)

```
Input: $\mathbf{X} \in \mathbb{R}^{N\times D}$, $\mathbf{H} \in \mathbb{R}^{N\times K}$, $\lambda, \lambda_1, \lambda_2, \lambda_3$ and $\eta$.
Initialize $\mathbf{W} = \mathbf{0}$, $\mathbf{\Omega}^{-1} = \mathbf{I}_K$, $\mathbf{\Sigma}_r^{-1} = \mathbf{I}_D$, $\mathbf{\Sigma}_c^{-1} = \mathbf{I}_K$ and $\mathbf{b} = \frac{1}{N}\mathbf{H}^\top\mathbf{1}$
while not converged do
  while not converged do
    Update $\mathbf{W}$ by $\mathbf{W} \leftarrow \mathbf{W} - \eta\nabla_{\mathbf{W}}\mathcal{L}_{\mathbf{W}}$
  end while
  Update $\mathbf{b}$ by $\hat{\mathbf{b}} = \frac{1}{N}(\mathbf{H} - \mathbf{X}\mathbf{W})^\top\mathbf{1}$
  Update $\mathbf{\Omega}^{-1}$ by (13)
  Update $\mathbf{\Sigma}_r^{-1}$ by (16)
  Update $\mathbf{\Sigma}_c^{-1}$ by (17)
end while
Output: $\mathbf{W}, \mathbf{b}, \mathbf{\Omega}^{-1}, \mathbf{\Sigma}_r^{-1}, \mathbf{\Sigma}_c^{-1}$
```

Algorithm 2 ICG Training (Unit2Pixel Stage)

```
Input: Training images $\mathbf{Y}$, conditions $\mathbf{H}$, hyperparameters $\alpha$ and $\beta$.
Initialize network parameters $\theta$ and $\phi$
while not converged do
  Update the generator by (18) and (19) :
    $\hat{\theta} = \arg\min_\theta[L_{AE} + \alpha D_{KL}(q_\phi(\mathbf{z}|\mathbf{y}_f, \mathbf{h})||p(\mathbf{z}))]$
  Update the encoder by (18) and (19) :
    $\hat{\phi} = \arg\min_\phi[L_{AE} + \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{h})||p(\mathbf{z}))$
                  $-\alpha D_{KL}(q_\phi(\mathbf{z}|\mathbf{y}_f, \mathbf{h})||p(\mathbf{z}))]$
end while
Output: $\hat{\theta}$ and $\hat{\phi}$
```

3) Update $\mathbf{\Omega}^{-1}$: Given $\mathbf{W}, \mathbf{b}, \mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$, the inverse covariance matrix $\mathbf{\Omega}^{-1}$ can be obtained by

$$\min_{\mathbf{\Omega}^{-1}} \operatorname{tr}((\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top)\mathbf{\Omega}^{-1}(\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top)^\top) - N\log|\mathbf{\Omega}^{-1}| + \lambda_2||\mathbf{\Omega}^{-1}||_1. \quad (12)$$

The above optimization problem can be solved by using the basic graphical lasso solver [45], i.e., $\hat{\mathbf{\Omega}}^{-1} = \text{GraphLasso}((1/N)(\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top)^\top(\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top),\ \lambda_2)$. However, its computational cost becomes prohibitive in high-dimensional output settings where K is large. Here, we employ a linear-time algorithm Thresh-max-det matrix completion (MDMC) proposed in [48] for large-scale sparse inverse covariance estimation

$$\hat{\mathbf{\Omega}}^{-1} = \text{Thresh-MDMC}(1/N \cdot (\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top)^\top \times (\mathbf{H} - \mathbf{X}\mathbf{W} - \mathbf{1}\mathbf{b}^\top),\ \lambda_2). \quad (13)$$

4) Update $\mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$: Given $\mathbf{W}, \mathbf{b}$ and $\mathbf{\Omega}^{-1}$, the inverse covariance matrices $\mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$ can be estimated by solving the following subproblems, respectively:

$$\min_{\mathbf{\Sigma}_r^{-1}} \lambda_1\operatorname{tr}\big(\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{\Sigma}_c^{-1}\mathbf{W}^\top\big) - K\log\big|\mathbf{\Sigma}_r^{-1}\big| + \lambda_3\big\|\mathbf{\Sigma}_r^{-1}\big\|_1 \quad (14)$$

$$\min_{\mathbf{\Sigma}_c^{-1}} \lambda_1\operatorname{tr}\big(\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{\Sigma}_c^{-1}\mathbf{W}^\top\big) - D\log\big|\mathbf{\Sigma}_c^{-1}\big| + \lambda_3\big\|\mathbf{\Sigma}_c^{-1}\big\|_1. \quad (15)$$

As in (13), we can estimate $\mathbf{\Sigma}_r^{-1}$ and $\mathbf{\Sigma}_c^{-1}$, as follows:

$$\hat{\mathbf{\Sigma}}_r^{-1} = \text{Thresh-MDMC}\big(1/K \cdot \mathbf{W}\mathbf{\Sigma}_c^{-1}\mathbf{W}^\top,\ \lambda_3/\lambda_1\big) \quad (16)$$

$$\hat{\mathbf{\Sigma}}_c^{-1} = \text{Thresh-MDMC}\big(1/D \cdot \mathbf{W}^\top\mathbf{\Sigma}_r^{-1}\mathbf{W},\ \lambda_3/\lambda_1\big). \quad (17)$$

The entire optimization procedure is summarized in Algorithm 1. Because each of the above subproblem is convex w.r.t. one variable, our algorithm can at least find a locally optimal solution by optimizing each subproblem alternatively.

B. Training ICG Model

The entire optimization procedure of ICG model is summarized in Algorithm 2. We alternately optimize (8) and (9), which correspond to updates to the generator and encoder, respectively. In each iteration, the generator parameter θ and encoder parameter ϕ can be optimized using the stochastic gradient variational Bayes (SGVB) [49] method. We assume $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the encoder $q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{h})$ is designed to output two individual variables, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and the approximated posterior $q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{h}) \sim \mathcal{N}(\boldsymbol{\mu}, \operatorname{diag}(\boldsymbol{\sigma}^2))$. Then, the KL-divergence term in (8) and (9) can be computed as

$$D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{h})||p(\mathbf{z})) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{d_z}\left(1 + \log\left(\boldsymbol{\sigma}_{ij}^2\right) - \boldsymbol{\mu}_{ij}^2 - \boldsymbol{\sigma}_{ij}^2\right) \quad (18)$$

where d_z denotes the dimension of $\mathbf{z}$. The image reconstruction error term L_{AE} in (8) and (9) is computed using the mean squared error (MSE) loss

$$L_{\text{AE}}(\mathbf{y}, \mathbf{y}_r) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\|\mathbf{y}_{r,ij} - \mathbf{y}_{ij}\|_2^2 \quad (19)$$

where $\mathbf{y}_r$ denotes the reconstructed image.

V. Experiments

A. Experimental Setup

1) Data Sets:

a) Vim-1: Vim-1 is a publicly available fMRI data set.[2] It contains two subjects' blood-oxygen-level dependent (BOLD) signals evoked by thousands of grayscale natural images (500 × 500 pixels) [20]. The fMRI voxels come from V1, V2, V3, V4, V3a, V3b, and lateral occipital (LO) visual brain areas. Each subject has five scan sessions, and each scan session consisted of seven runs, where five runs were used for model training and the rest two runs for model testing. See [20] for more details about Vim-1.

b) FaceBold: Our newly collected fMRI data set,[3] which contains six subjects' BOLD signals evoked by 800 grayscale face stimuli. The face stimuli (330 × 380 pixels) come from several public emotion data sets such as the Radboud Faces Database [50]. During the acquisition, we recorded the BOLD responses (repetition time (TR) = 2 s, voxel size = $3 \times 3 \times 4$ mm^3, whole-brain coverage) of six subjects (three female

[2] Data are available at https://crcns.org/datasets/vc/vim-1

[3] Data are available at https://figshare.com/articles/dataset/FaceBold/13019966

TABLE II
DETAILS OF THE DATA SETS USED IN OUR EXPERIMENTS

Dataset	Training (N)	Test	Voxels (D)	Pixels	Augmentation
Vim-1	1750	120	8428	128×128	ImageNet-1k
FaceBold	720	80	5000	128×128	CelebA

and three males, 20–30 years old) as they were fixating on a small dot superimposed on the stimuli (15° × 15°). Each face was presented at a frequency of 5 Hz for 2 s, followed by a gray background for 6 s. Each subject has two scan sessions. The first session consisted of five runs, which were used for model training totally. The second session also consisted of five runs, but only four runs were used for model training and the rest one for model testing. For each run, we randomly select 80 different face stimuli and ensure that the face stimuli in each run do not overlap.

c) ImageNet-1k: The ImageNet-1k [51] data set contains approximately 1 280 000 natural images. Before using it to augment the training set, we do some preprocessing on it. First, the image size of ImageNet-1k was downsampled to 128 × 128 using the method proposed in [52]. Second, similar to [20], we converted all the images to grayscale and enhanced their contrast. Finally, to make images look like the ones in Vim-1, we finally applied the circular mask [20] to them. The preprocessed ImageNet-1k images together with the downsampled (128 × 128) Vim-1 images were used to train our ICG model.

d) CelebA: The CelebA [53] data set[4] contains approximately 203 000 face images. We use the aligned and cropped version (i.e., the "img_align_celeba.zip" file) in our experiments. The face images in CelebA were center-cropped to 128 × 128 to align them with the images in FaceBold. Finally, all face images were converted to gray scale to augment the training set.

We summarized the properties of these data sets in Table II.

2) Compared Methods: In *Voxel2Unit*, we compare our SMR with 1) **SLR**: [6]; 2) **BCCA**: Bayesian canonical correlation analysis [54]; 3) various special cases of our SMR model, e.g., **SMR-O**: SMR with only output structure, i.e., $\mathbf{\Sigma}_r^{-1} = \mathbf{I}$ and $\mathbf{\Sigma}_c^{-1} = \mathbf{I}$ [24]; **SMR-IT**: SMR with only input and task structures, i.e., $\mathbf{\Omega}^{-1} = \mathbf{I}$ [27]. In *Unit2Pixel*, we compare our ICG model with 1) **CVAE** [5], [10]; 2) **CGAN**: [4], [42]; 3) **Grad-TV**: gradient-based optimization with a TV regularizer [7]; 4) **De-CNN** [9]; 5) **VAEGAN** [55].

3) Parameter Settings: Depending on whether sparse constraints are imposed on the inverse covariance matrices, we consider the nonsparse and sparse two variants of SMR. The hyperparameter $\lambda = 0.001$ for both cases. For nonsparse SMR, we fix $\lambda_2 = \lambda_3 = 10^{-6}$, and λ_1 is selected using five-fold cross-validation within the range $[10^{-5}, 10^3]$. For sparse SMR, we use the same value of λ_1 that was selected for nonsparse case, and only λ_2 and λ_3 are selected by cross-validation. Due to the large number of CNN units, it is difficult to put all of them into memory for multioutput regression training. To overcome this problem, we use segment-training methods. Specifically, we first divide all CNN units into many segments in order, so that the number of CNN units in each segment was about 40 000. Then, we train the SMR on each segment of data.

For the proposed ICG model, the top 5000 decodable CNN units (according to the rank of each unit's decodability) were treated as the condition, and we set $\{\alpha = 0.5, \beta = 1\}$ to combine the advantages of both CVAE and CGAN. We use ICG model with $\{\alpha = 0, \beta = 1\}$ to implement CVAE and ICG model with $\{\alpha = 1, \beta = 0\}$ to implement CGAN. The latent variable $\mathbf{z}$ is randomly drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, with the dimension set to 512 and 256 on the Vim-1 and FaceBold data sets, respectively. For optimizing the proposed ICG model, we used the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [56] with a batch size of 32 and a fixed learning rate 0.0001.

B. Experimental Results

In the test phase, we first need to predict the CNN features from the brain activity and then invert the decoded CNN features back to the visual images.

1) CNN Feature Decoding: We first perform Voxel2Unit analysis to decode the AlexNet [43] features from the fMRI activities. The experimental results on the Vim-1 and FaceBold data sets are displayed in Table III. From Table III, we can find that SMR consistently achieves lower normalized MSEs (NMSEs) on the test set among all the methods. First, by comparing our SMR against the baseline methods, we see that the performance of the sparse SMR are significantly ($p < 0.05$) better than SLR and BCCA. The biggest advantage of SMR is that it can leverage the covariance structures over fMRI voxels, regression tasks and CNN units simultaneously, whereas SLR and BCCA cannot. Second, the results of comparing SMR against its six special cases (SMR-T, SMR-OT, etc.) show that the averaged NMSEs are increasing as we impose more structural constraints, although the results were not statistically significant ($p > 0.05$). Due to the learned structural constraints also have a certain interpretability, it is a reasonable way to apply them in the stage of CNN feature decoding. Finally, we also see that the sparse performance is better than nonsparse performance in most cases. This shows that explicitly encouraging zero entries in the inverse covariance matrices leads to better estimations of the structures. More comparisons on each individual subject can be found in Appendix C.

The experiments of Table III are conducted on the default test data sets. To assess whether the effectiveness of our approach depends on data splits, we perform fivefold cross-validation over all of the fMRI runs. In fivefold cross-validation, only one of the five folds (iterate through each five different folds) is used as the test set while the rest is used for training. The results are listed in Table IV. It can be seen that the performance difference of SMR under different folds is small.

2) Analyze the Performance Layer by Layer: To explore the value of different layers in neural decoding tasks, we analyzed the prediction accuracy of each AlexNet layer on the Vim-1 data set (see Fig. 4). Here, we use the Pearson correlation

[4] Data are available at http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

TABLE III

AVERAGE NMSE ACROSS ALEXNET LAYERS, UNITS AND SUBJECTS WITH MEAN ± STD (t-VALUE, p-VALUE) FORMAT. ● INDICATES THAT SMR IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHOD ($p < 0.05$), WHERE THE p-VALUES HAVE BEEN CORRECTED WITH BONFERRONI METHOD FOR MULTIPLE COMPARISONS

Method	Vim-1		FaceBold	
	Non-sparse	Sparse	Non-sparse	Sparse
SLR	-	.636 ± .044 (4.06, 0.035)●	-	.721 ± .035 (4.17, 0.024)●
BCCA	-	.687 ± .035 (4.23, 0.011)●	-	.789 ± .039 (4.31, 0.007)●
SMR-T	.589 ± .035 (2.51, 0.238)	.582 ± .033 (2.43, 0.308)	.714 ± .041 (3.33, 0.084)	.694 ± .043 (3.45, 0.011)●
SMR-I	.585 ± .034 (2.36, 0.322)	.579 ± .037 (2.42, 0.322)	.716 ± .038 (3.34, 0.098)	.675 ± .041 (3.27, 0.112)
SMR-O	.583 ± .041 (2.22, 0.406)	.581 ± .045 (2.40, 0.35)	.721 ± .045 (3.35, 0.07)	.685 ± .044 (3.31, 0.098)
SMR-OT	.569 ± .038 (2.11, 0.518)	.568 ± .038 (2.28, 0.434)	.686 ± .043 (1.88, 0.77)	.679 ± .036 (2.42, 0.518)
SMR-IT	.561 ± .039 (1.90, 0.77)	.566 ± .044 (2.31, 0.49)	.692 ± .046 (1.91, 0.672)	.668 ± .040 (1.86, 0.798)
SMR-IO	.568 ± .042 (2.04, 0.56)	.560 ± .042 (1.77, 0.868)	.684 ± .039 (1.85, 0.826)	.670 ± .043 (1.82, 0.854)
SMR	**.558** ± .043	**.552** ± .039	**.676** ± .042	**.664** ± .039

TABLE IV

NMSE OF THE PROPOSED SMR METHOD ON EACH FOLDS. RESULTS ARE AVERAGE ACROSS CNN (ALEXNET) LAYERS, UNITS, AND SUBJECTS

SMR	Dataset	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average
Non-sparse	Vim-1	.549	.563	.576	.552	.545	.557 ± .011
	FaceBold	.672	.677	.661	.658	.651	.664 ± .010
Sparse	Vim-1	.545	.557	.573	.546	.538	.551 ± .012
	FaceBold	.669	.672	.656	.654	.648	.660 ± .009

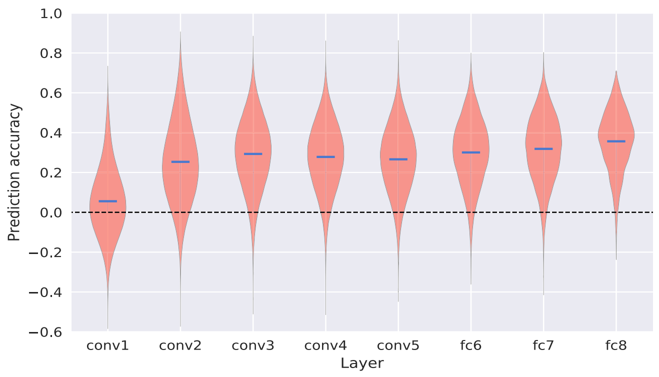

Fig. 4. Decoding accuracy for individual AlexNet layer. The results have been averaged over subjects. The blue bars represent the average prediction accuracy of all units in each layer.

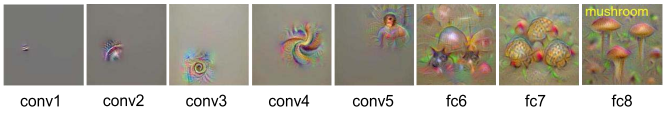

Fig. 5. Examples of preferred images for randomly selected units. Adapted from [6].

coefficient (PCC) between the true CNN feature and the predicted ones (by using our SMR model) as a metric. The results show that the decoding effect of different layers is obviously different, and the differences of distinct CNN units in the same layer are also great. Furthermore, the generalization performance of the deep features (conv3 to fc8) is better than that of the shallow features (conv1, conv2). This may be because deep layers have larger receptive fields and richer structural information (as shown in Fig. 5). Logically, selecting CNN features with higher prediction accuracy for image reconstruction will be beneficial to improve the generalization performance of the model.

3) Relationship Between CNN Layer and ROI of Brain: As shown in Fig. 5, the feature complexity in a forward CNN is hierarchically organized. Previous studies [34] have shown that the visual pathway of the human brain is also hierarchical in feature processing. In order to explore the relationship between different CNN layer and brain regions-of-interest (ROI), we visualized the relative contribution of different ROIs to the prediction of features at different CNN layers by using the regression weights of the trained SMR model. The results are shown in Fig. 6, from which we can see that the primary visual regions (V1, V2) make greater contribution to the decoding of the shallower CNN layer and smaller contribution to the decoding of the deeper CNN layer. On the contrary, the decoding contribution of the downstream visual regions (V4, LO) to the shallow CNN layer is relatively small, while the decoding contribution to the deep CNN layer is relatively large. These experimental results show that there is homology between deep CNN and human visual pathway in visual feature processing, and also confirm the rationality of neural decoding using CNN intermediate features.

4) Performance by Different CNN Architectures: The above decoding results are obtained using the intermediate features of AlexNet [43]. Compared to more modern CNN architectures [57], [58], the diversity of AlexNet's hierarchical features is not rich enough. By contrast, due to the large difference in the receptive field range of units at different depth, the hierarchical features of ResNet have richer expression ability. It is assumed that the use of this rich hierarchical feature information may be more conducive to neural decoding. In Fig. 7, we show the decoding results of VGG16 [57] and ResNet18 [58] on the Vim-1 test set. Surprisingly, we do not find that the choice of CNN architecture has a significant effect on neural decoding performance. However, the computational time of decoding is greatly increased, since VGG16/ResNet18 has many more intermediate features than AlexNet.

5) Differences Between Subjects: To evaluate the decoding performance differences between subjects, we compared the decoding results of two subjects in the vim-1 data set unit by unit, and the scatter plots are shown in Fig. 8. The vertical and horizontal coordinates of each unit represent the decoding accuracy of Subject-1 and Subject-2, respectively. The PCCs are displayed in the lower right corner of each panel. We see that, points are densely distributed along the diagonal, and the

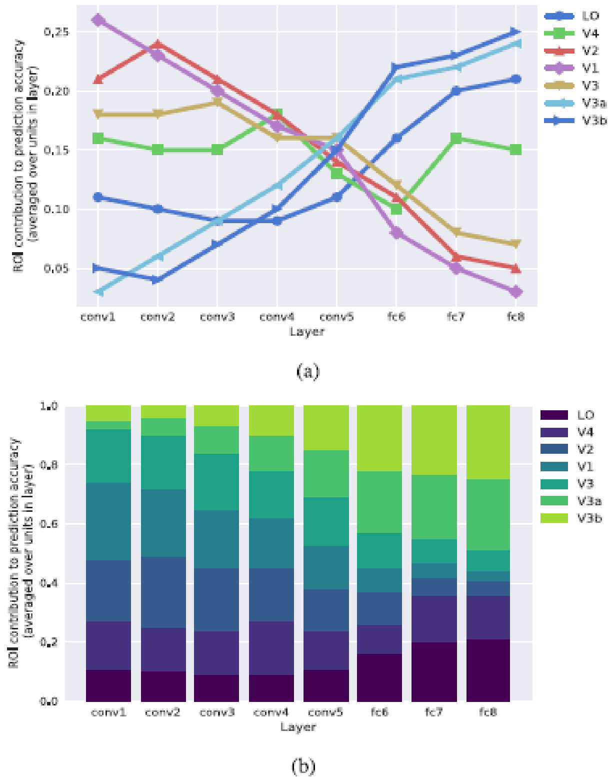

Fig. 6. Contributions of the brain ROIs to CNN layer predictions. The result for each layer is the average of all units in that layer.

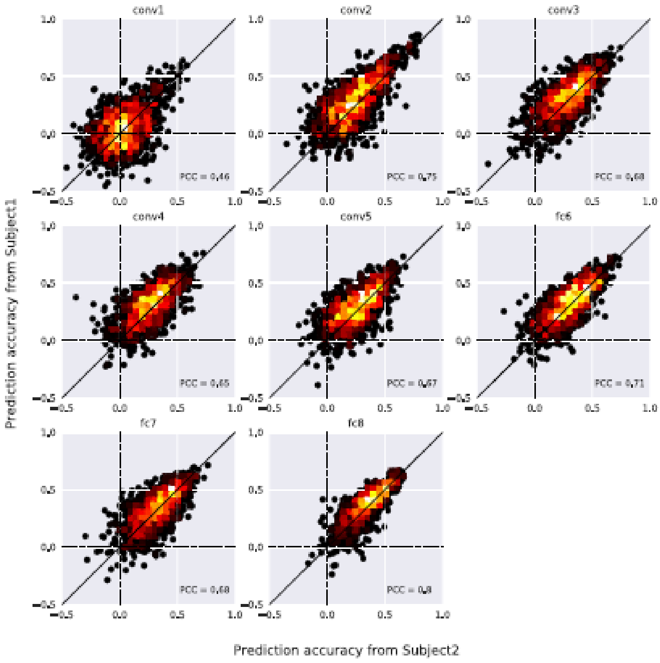

Fig. 8. Decoding accuracy for each layer of AlexNet using different subjects' fMRI data of the Vim-1 data set. Each dot represents a CNN unit. The vertical and horizontal coordinates of each dot represent the decoding accuracy of that unit of Subject-1 and Subject-2, respectively. The brighter the place, the denser the dot. For each layer, we randomly selected 1000 units for better visualization.

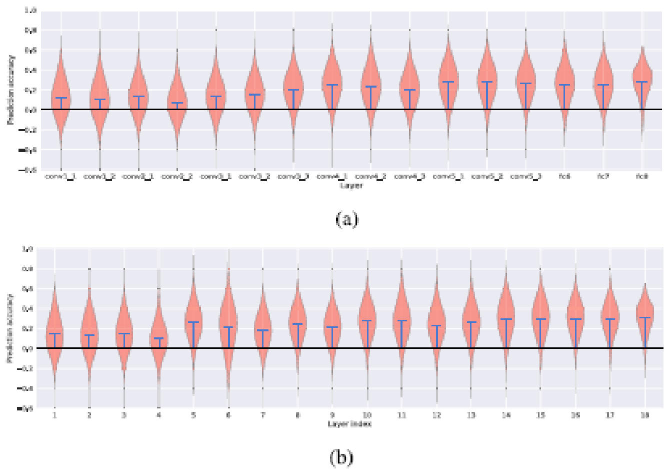

Fig. 7. Decoding performance for each layer of VGG16 and ResNet18. The results have been averaged over subjects. The blue bars represent the average prediction accuracy of all units in each layer. (a) VGG16. (b) ResNet18.

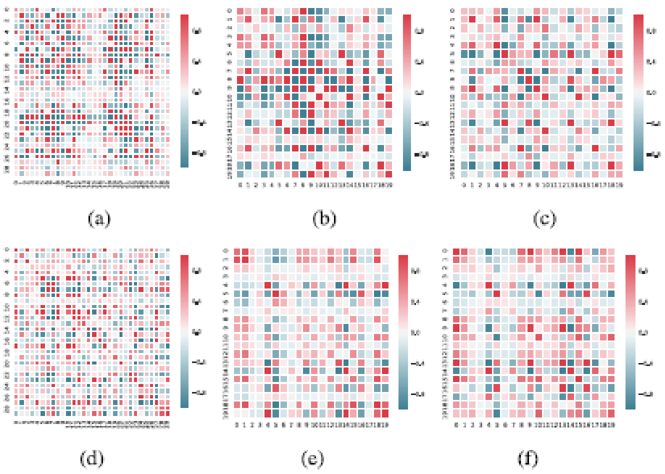

Fig. 9. Visualization of the inverse covariance matrices learned by SMR on the Vim-1 (top row: Subject-1; bottom row: Subject-2). (a) Input structure $\boldsymbol{\Sigma}_r^{-1}$. (b) Task structure $\boldsymbol{\Sigma}_c^{-1}$. (c) Output structure $\boldsymbol{\Omega}^{-1}$. (d) Input structure $\boldsymbol{\Sigma}_r^{-1}$. (e) Task structure $\boldsymbol{\Sigma}_c^{-1}$. (f) Output structure $\boldsymbol{\Omega}^{-1}$.

PCC values are positive for all layers. This indicates that the CNN feature decoding from the brain using our SMR model was highly consistent across subjects.

6) Covariance Structures Visualization: Our SMR model can simultaneously leverage the covariance structures underlying the brain voxels, the CNN units and the prediction tasks to improve the decoding accuracy and interpretability. In Fig. 9, we visualized the sparse inverse covariance matrices after training. For better observation, 30 voxels from V1 and 20 units from conv1 of AlexNet were selected to display. As we can see, the inverse covariance matrices successfully exhibit the underlying structures of voxels, tasks, and units. For example, in Subject-1, the voxels #20 − −#22 are highly correlated with each other, and units #0− −#4 also have strong interactions with each other.

7) Convergence Studies: We analyze the convergence of our proposed SMR method by recording the changes of objective function value and the fitting error during optimization. The `log` value of the objective function [given by (6)] and the average MSE on the Vim-1 data set are shown in Fig. 10.

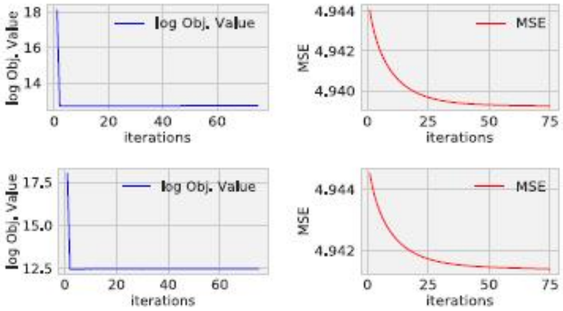

Fig. 10. Convergence properties of the proposed SMR model on the Vim-1 data set (top row: Subject-1; bottom row: Subject-2).

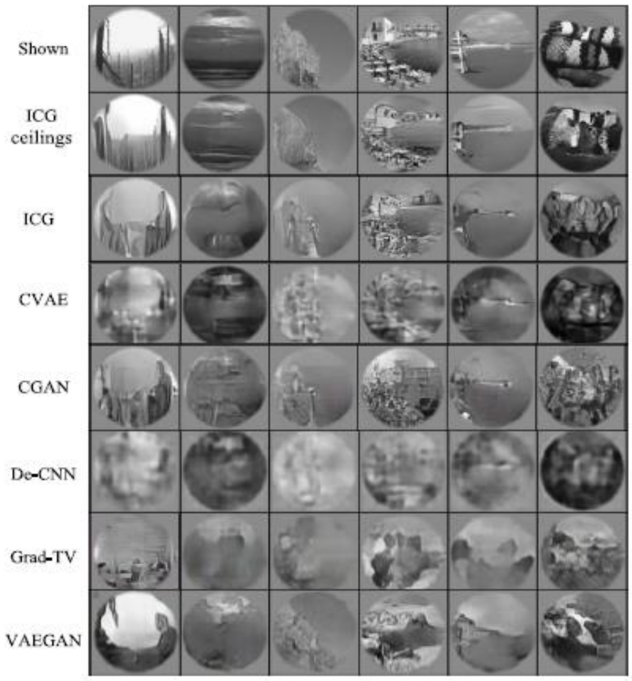

Fig. 11. Examples of reconstructed natural images on the Vim-1.

From these curves, we can observe that the process of CNN feature decoding using our SMR method can converge after about 50 iterations.

8) Perceived Image Reconstruction: We finally conduct the Unit2Pixel experiments to reconstruct the perceived images from the decoded CNN features (we only use the top 5000 decodable CNN units of AlexNet). Figs. 11 and 12 show several representative examples of the test stimuli and their reconstructions on both data sets (additional results are given in Appendix B). ICG ceilings refer to the best result that can be obtained by using our ICG model for neural decoding reconstruction, and they are obtained by using the true CNN features in ICG model. CVAE, CGAN, De-CNN, and Grad-TV also use the hierarchical CNN features predicted by our SMR, while VAEGAN only the predicted bottleneck layer features.

Note that the fourth and fifth rows study the ablation effects of ICG–both CVAE and CGAN are the special cases of ICG. When we set $\alpha = 0, \beta = 1$, the ICG degrades into the CVAE and when we set $\alpha = 1, \beta = 0$, it degrades into the CGAN. Through visual observation of the reconstruction results of ICG model, we find that they are consistent with the given stimulus image in terms of contour, texture, and some semantic features. Note that both the CGAN and our ICG models are based on adversarial learning. Although the difference between their image reconstruction quality is not huge, our ICG model can leverage the characteristics of CVAE to improve its stability during training. Although VAEGAN's reconstruction images looks as sharp as CGAN and our ICG, several of its reconstructions don't respect the original images very well in terms of shape and details. We attribute this to the fact that only the bottleneck layer features are used in VAEGAN to reconstruct the images.

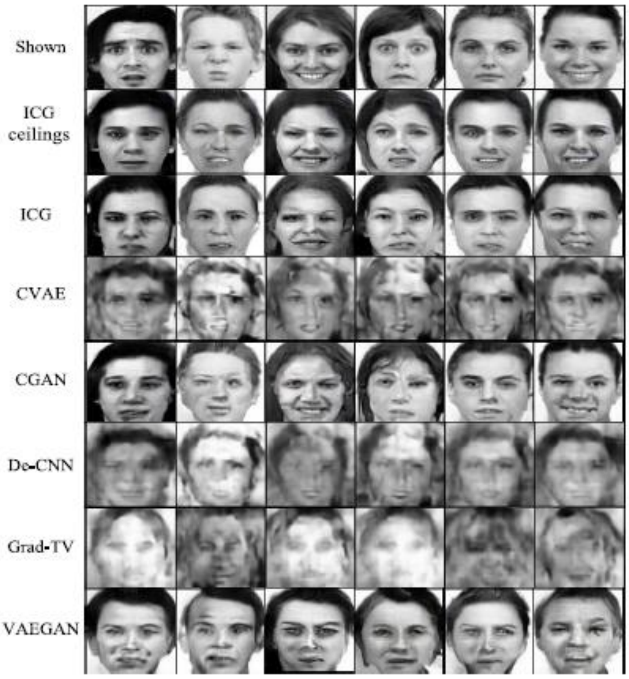

Fig. 12. Examples of reconstructed human faces on the FaceBold.

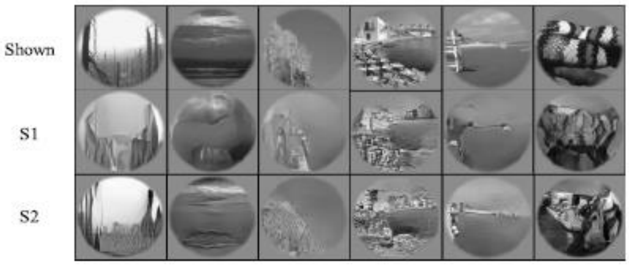

Fig. 13. Examples of the ICG reconstructed natural images on the two subjects (S1 to S2) of Vim-1 data set.

The image reconstruction accuracy of different competitors on the Vim-1 and FaceBold data sets are shown in Table V. Specifically, we used PCC and structural similarity index (SSIM) metrics to evaluate their performance. Each entry in Table V denotes the ratio of the reconstruction accuracies of the predicted and the ground-truth CNN features. From Table V, we can observe that the performance of our ICG model are significantly ($p < 0.05$) better than Grad-TV and

TABLE V

RECONSTRUCTION ACCURACY MEASURED BY PCC AND SSIM WITH MEAN ± STD (t-VALUE, p-VALUE) FORMAT. ● INDICATES THAT ICG IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHOD ($p < 0.05$), WHERE THE p-VALUES HAVE BEEN CORRECTED WITH BONFERRONI METHOD FOR MULTIPLE COMPARISONS

Method	Vim-1		FaceBold	
	PCC	SSIM	PCC	SSIM
Grad-TV	.263 ± .055 (3.62, 0.03)●	.350 ± .039 (4.11, 0.02)●	.374 ± .051 (3.87, 0.031)●	.432 ± .044 (4.06, 0.035)●
De-CNN	.458 ± .044 (3.57, 0.035)●	.545 ± .027 (3.68, 0.022)●	.548 ± .046 (3.89, 0.038)●	.755 ± .037 (3.65, 0.065)
CVAE	.476 ± .045 (3.25, 0.1)	.594 ± .030 (3.08, 0.105)	.576 ± .048 (3.13, 0.095)	.794 ± .031 (3.31, 0.105)
CGAN	.493 ± .043 (2.95, 0.16)	.625 ± .029 (2.42, 0.215)	.593 ± .045 (2.27, 0.195)	.825 ± .027 (2.88, 0.185)
VAEGAN	.471 ± .047 (3.08, 0.115)	.582 ± .033 (2.76, 0.18)	.581 ± .049 (2.44, 0.12)	.786 ± .033 (2.86, 0.13)
ICG	**.552** ± .044	**.672** ± .024	**.652** ± .042	**.872** ± .025

TABLE VI

MEAN ± STD (t-VALUE, BONFERRONI CORRECTED p-VALUE) RESULTS ON THE BOTH DATA SETS. SIGNIFICANT THRESHOLD IS $p = 0.05$

Method	Vim-1			
	Subject-1		Subject-2	
	Non-sparse	Sparse	Non-sparse	Sparse
SLR	-	.629 ± .041 (3.96, 0.025)●	-	.640 ± .038 (4.11, 0.017)●
BCCA	-	.682 ± .033 (4.35, 0.01)●	-	.688 ± .033 (4.17, 0.009)●
SMR-T	.572 ± .031 (2.44, 0.196)	.570 ± .030 (2.51, 0.28)	.593 ± .033 (2.43, 0.266)	.588 ± .037 (2.45, 0.294)
SMR-I	.584 ± .030 (2.38, 0.252)	.580 ± .035 (2.44, 0.294)	.587 ± .029 (2.45, 0.238)	.582 ± .032 (2.33, 0.28)
SMR-O	.579 ± .031 (2.26, 0.35)	.575 ± .042 (2.43, 0.308)	.588 ± .038 (2.37, 0.308)	.587 ± .042 (2.17, 0.336)
SMR-OT	.561 ± .036 (1.87, 0.86)	.558 ± .037 (2.02, 0.658)	.574 ± .035 (2.16, 0.462)	.570 ± .035 (2.32, 0.392)
SMR-IT	.560 ± .036 (1.91, 0.84)	.562 ± .041 (2.27, 0.63)	.562 ± .038 (1.87, 0.924)	.563 ± .042 (2.11, 0.574)
SMR-IO	.564 ± .037 (2.00, 0.588)	.561 ± .040 (1.83, 0.812)	.570 ± .031 (2.56, 0.42)	.566 ± .045 (1.75, 0.84)
SMR	**.552** ± .040	**.549** ± .034	**.560** ± .036	**.553** ± .036
	FaceBold			
	Subject-1		Subject-2	
SLR	-	.715 ± .032 (4.23, 0.022)●	-	.718 ± .037 (4.11, 0.028)●
BCCA	-	.793 ± .035 (4.29, 0.011)●	-	.782 ± .031 (4.25, 0.009)●
SMR-T	.711 ± .039 (3.57, 0.098)	.699 ± .029 (3.72, 0.084)	.722 ± .033 (3.98, 0.126)	.702 ± .038 (3.28, 0.07)
SMR-I	.719 ± .037 (3.42, 0.084)	.683 ± .033 (3.94, 0.098)	.725 ± .036 (4.01, 0.084)	.689 ± .040 (3.68, 0.042)●
SMR-O	.715 ± .041 (3.66, 0.098)	.677 ± .047 (3.45, 0.07)	.717 ± .031 (3.41, 0.112)	.687 ± .037 (3.37, 0.084)
SMR-OT	.692 ± .037 (1.87, 0.672)	.682 ± .034 (2.54, 0.406)	.682 ± .040 (1.76, 0.994)	.674 ± .035 (1.92, 0.896)
SMR-IT	.689 ± .036 (1.89, 0.63)	.669 ± .038 (1.91, 0.708)	.685 ± .041 (1.94, 0.574)	.676 ± .040 (1.88, 0.882)
SMR-IO	.679 ± .040 (1.85, 0.77)	.676 ± .032 (1.96, 0.82)	.683 ± .039 (1.84, 0.798)	.675 ± .043 (1.96, 0.77)
SMR	**.668** ± .036	**.665** ± .034	**.679** ± .038	**.671** ± .039
	Subject-3		Subject-4	
SLR	-	.726 ± .038 (4.51, 0.042)●	-	.712 ± .032 (4.22, 0.031)●
BCCA	-	.794 ± .034 (4.18, 0.008)●	-	.783 ± .034 (4.13, 0.01)●
SMR-T	.706 ± .032 (3.36, 0.112)	.699 ± .034 (3.82, 0.084)	.721 ± .031 (3.21, 0.07)	.691 ± .035 (3.25, 0.042)●
SMR-I	.712 ± .028 (3.98, 0.084)	.682 ± .026 (3.34, 0.084)	.719 ± .036 (3.46, 0.084)	.688 ± .037 (3.57, 0.084)
SMR-O	.705 ± .037 (3.78, 0.056)	.680 ± .030 (3.64, 0.042)●	.718 ± .037 (3.24, 0.084)	.680 ± .031 (3.43, 0.098)
SMR-OT	.685 ± .030 (1.88, 0.728)	.675 ± .041 (2.47, 0.476)	.687 ± .029 (1.85, 0.882)	.677 ± .032 (2.64, 0.448)
SMR-IT	.688 ± .042 (1.95, 0.602)	.672 ± .032 (2.62, 0.448)	.689 ± .030 (1.87, 0.77)	.668 ± .029 (1.87, 0.728)
SMR-IO	.680 ± .034 (1.82, 0.938)	.665 ± .040 (1.85, 0.77)	.696 ± .034 (1.94, 0.532)	.665 ± .031 (1.81, 0.924)
SMR	**.672** ± .028	**.658** ± .033	**.683** ± .031	**.662** ± .028
	Subject-5		Subject-6	
SLR	-	.718 ± .035 (4.35, 0.035)●	-	.722 ± .036 (4.26, 0.026)●
BCCA	-	.784 ± .039 (4.27, 0.006)●	-	.787 ± .038 (4.45, 0.008)●
SMR-T	.712 ± .038 (3.42, 0.098)	.694 ± .038 (3.65, 0.112)	.715 ± .040 (3.52, 0.126)	.696 ± .028 (3.36, 0.024)●
SMR-I	.706 ± .032 (3.67, 0.07)	.675 ± .041 (3.45, 0.084)	.713 ± .039 (3.36, 0.07)	.679 ± .043 (3.58, 0.098)
SMR-O	.724 ± .041 (3.52, 0.126)	.685 ± .040 (3.72, 0.084)	.723 ± .024 (3.71, 0.07)	.683 ± .040 (3.61, 0.056)
SMR-OT	.695 ± .036 (1.98, 0.532)	.669 ± .042 (1.88, 0.826)	.688 ± .042 (1.78, 0.854)	.669 ± .031 (1.81, 0.826)
SMR-IT	.681 ± .033 (1.79, 0.728)	.666 ± .036 (1.98, 0.728)	.690 ± .034 (1.98, 0.574)	.678 ± .037 (2.16, 0.658)
SMR-IO	.679 ± .029 (1.89, 0.798)	.674 ± .035 (2.39, 0.574)	.681 ± .038 (1.94, 0.84)	.672 ± .028 (1.79, 0.868)
SMR	**.675** ± .035	**.663** ± .032	**.674** ± .030	**.667** ± .033

De-CNN in most cases. Compared with the other baselines (CVAE, CGAN, and VAEGAN), although the results were not statistically significant ($p > 0.05$), our approach is more flexible in the learning framework and it produce higher visual quality reconstructions. In addition, we show the examples of ICG reconstructed results for each subject (S1–S2 for the Vim-1 and S1–S6 for the FaceBold) in Figs. 13 and 14 based on the predicted CNN features of each subject using our SMR model.

VI. CONCLUSION

To tackle the perceived image reconstruction problem, we have proposed a structured neural decoding method based on multitask transfer learning of DNN representations. Our method involves two cascaded stages. In the first stage, we use matrix-variable Gaussian prior established an SMR model to decode the multivariate fMRI data to the CNN features. The SMR can simultaneously take into account the covariance structures between the data dimensions and the regression tasks. In the second stage, we built an ICG model, which can be trained easily to produce high-fidelity image reconstructions. We studied the relationship between different feature layers of three CNN architectures (AlexNet, VGG16, ResNet18) and different brain visual areas, and found that there is a homology between computer and human vision. The experimental results demonstrate that our method can

TABLE VII

MEAN ± STD (t-VALUE, BONFERRONI CORRECTED p-VALUE) RESULTS ON THE BOTH DATA SETS. SIGNIFICANT THRESHOLD IS $p = 0.05$

Method	Vim-1			
	Subject-1		Subject-2	
	PCC	SSIM	PCC	SSIM
Grad-TV	.258 ± .043 (3.42, 0.028)●	.347 ± .036 (4.11, 0.021)●	.274 ± .047 (3.68, 0.033)●	.353 ± .034 (4.01, 0.039)●
De-CNN	.436 ± .038 (3.56, 0.036)●	.528 ± .028 (3.78, 0.017)●	.463 ± .038 (3.61, 0.041)●	.544 ± .025 (3.58, 0.023)●
CVAE	.472 ± .029 (2.93, 0.13)	.583 ± .031 (2.78, 0.14)	.485 ± .034 (2.89, 0.16)	.597 ± .023 (2.78, 0.145)
CGAN	.498 ± .031 (2.75, 0.175)	.589 ± .029 (2.45, 0.205)	.473 ± .031 (2.76, 0.175)	.632 ± .021 (2.52, 0.185)
VAEGAN	.465 ± .032 (2.87, 0.155)	.572 ± .037 (2.56, 0.19)	.492 ± .030 (3.01, 0.13)	.594 ± .027 (2.36, 0.195)
ICG	**.545** ± .032	**.664** ± .024	**.582** ± .033	**.682** ± .021
	FaceBold			
	Subject-1		Subject-2	
Grad-TV	.382 ± .047 (3.77, 0.023)●	.471 ± .039 (3.96, 0.027)●	.386 ± .040 (3.67, 0.023)●	.441 ± .039 (3.96, 0.029)●
De-CNN	.518 ± .039 (3.81, 0.028)●	.741 ± .031 (3.55, 0.034)●	.552 ± .042 (3.79, 0.031)●	.713 ± .033 (3.71, 0.036)●
CVAE	.536 ± .037 (3.03, 0.115)	.764 ± .027 (3.01, 0.135)	.582 ± .041 (2.93, 0.135)	.738 ± .028 (3.13, 0.135)
CGAN	.573 ± .035 (2.37, 0.175)	.805 ± .022 (2.36, 0.17)	.586 ± .039 (2.36, 0.175)	.787 ± .022 (2.41, 0.16)
VAEGAN	.575 ± .041 (2.48, 0.110)	.765 ± .028 (2.66, 0.16)	.563 ± .043 (2.24, 0.140)	.731 ± .029 (2.75, 0.155)
ICG	**.681** ± .037	**.883** ± .033	**.647** ± .036	**.864** ± .031
	Subject-3		Subject-4	
Grad-TV	.368 ± .044 (3.76, 0.018)●	.425 ± .041 (4.15, 0.029)●	.393 ± .043 (4.02, 0.023)●	.443 ± .040 (4.12, 0.028)●
De-CNN	.521 ± .042 (3.55, 0.025)●	.737 ± .039 (3.48, 0.035)●	.562 ± .041 (3.23, 0.022)●	.782 ± .034 (3.89, 0.042)●
CVAE	.549 ± .041 (3.05, 0.11)	.768 ± .036 (3.34, 0.046)●	.588 ± .044 (3.03, 0.105)	.812 ± .035 (3.23, 0.13)
CGAN	.587 ± .038 (2.31, 0.16)	.799 ± .032 (2.12, 0.235)	.612 ± .038 (2.17, 0.175)	.833 ± .032 (3.54, 0.135)
VAEGAN	.553 ± .046 (2.36, 0.145)	.773 ± .040 (2.93, 0.105)	.595 ± .035 (2.28, 0.17)	.807 ± .034 (2.66, 0.175)
ICG	**.643** ± .042	**.860** ± .025	**.675** ± .042	**.887** ± .030
	Subject-5		Subject-6	
Grad-TV	.363 ± .043 (3.87, 0.032)●	.424 ± .038 (4.23, 0.029)●	.378 ± .035 (4.13, 0.022)●	.424 ± .037 (4.06, 0.026)●
De-CNN	.535 ± .041 (3.89, 0.023)●	.747 ± .034 (3.57, 0.05)	.553 ± .034 (3.69, 0.03)●	.749 ± .034 (3.03, 0.065)
CVAE	.569 ± .044 (3.13, 0.095)	.774 ± .038 (3.44, 0.065)	.587 ± .044 (3.54, 0.06)	.774 ± .035 (2.78, 0.105)
CGAN	.577 ± .038 (2.27, 0.195)	.806 ± .036 (2.72, 0.11)	.591 ± .039 (2.41, 0.115)	.816 ± .034 (2.67, 0.175)
VAEGAN	.565 ± .033 (2.44, 0.12)	.768 ± .040 (2.68, 0.1)	.580 ± .041 (2.57, 0.095)	.764 ± .029 (2.82, 0.145)
ICG	**.636** ± .034	**.857** ± .029	**.639** ± .037	**.869** ± .027

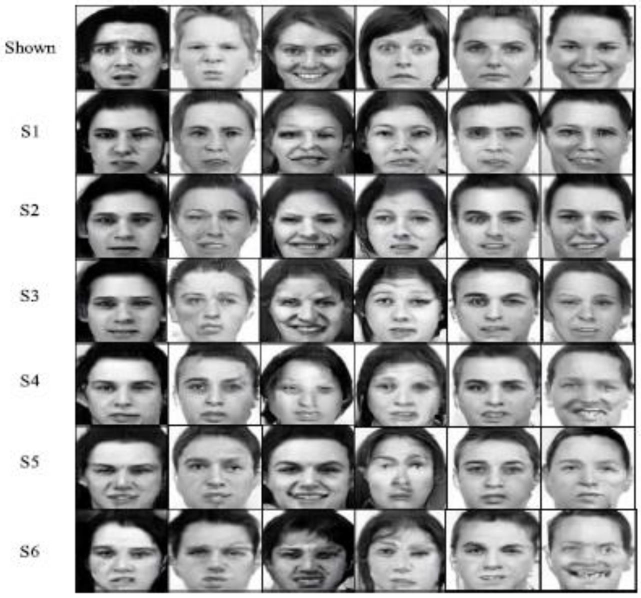

Fig. 14. Examples of the ICG reconstructed human faces on the six subjects (S1 to S6) of FaceBold data set.

accurately reconstruct the perceived natural images and human faces from brain activity.

APPENDIX A
PROOF OF PROPOSITION 1

First, we should know that the following conclusions hold true.

Fact 1: Let $\mathbf{A}$ be a matrix. Then $||\mathbf{A}||_F = ||\text{vec}(\mathbf{A})||_2$.

Fact 2: Let $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$, $\mathbf{B} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{C} \in \mathbb{R}^{n_2 \times m_2}$. Then $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\text{vec}(\mathbf{B})$.

Fact 3: Let $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}, \mathbf{B} \in \mathbb{R}^{n_1 \times p_1}$, $\mathbf{C} \in \mathbb{R}^{m_2 \times n_2}$ and $\mathbf{D} \in \mathbb{R}^{n_2 \times p_2}$. Then $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AB}) \otimes (\mathbf{CD})$.

Then, the objective function about $\mathbf{W}$ can be rewritten as

$$\begin{aligned}
&\mathcal{L}_{\mathbf{W}}\\
&= \text{tr}((\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-1}(\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)^\top)\\
&\quad + \lambda\text{tr}(\mathbf{WW}^\top) + \lambda_1\text{tr}\left(\boldsymbol{\Sigma}_r^{-1}\mathbf{W}\boldsymbol{\Sigma}_c^{-1}\mathbf{W}^\top\right)\\
&= \left\|(\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right\|_F^2 + \lambda\|\mathbf{W}\|_F^2\\
&\quad + \lambda_1\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{W}\boldsymbol{\Sigma}_c^{-\frac{1}{2}}\right\|_F^2\\
&= \left\|\text{vec}\left[(\mathbf{H} - \mathbf{XW} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]\right\|_2^2 + \lambda\|\text{vec}(\mathbf{W})\|_2^2\\
&\quad + \lambda_1\left\|\text{vec}\left(\boldsymbol{\Sigma}_r^{-\frac{1}{2}}\mathbf{W}\boldsymbol{\Sigma}_c^{-\frac{1}{2}}\right)\right\|_2^2\\
&= \left\|\text{vec}\left[(\mathbf{H} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right] - (\boldsymbol{\Omega}^{-\frac{1}{2}} \otimes \mathbf{X})\text{vec}(\mathbf{W})\right\|_2^2\\
&\quad + \lambda\|\text{vec}(\mathbf{W})\|_2^2 + \lambda_1\left\|\left(\boldsymbol{\Sigma}_c^{-\frac{1}{2}} \otimes \boldsymbol{\Sigma}_r^{-\frac{1}{2}}\right)\text{vec}(\mathbf{W})\right\|_2^2\\
&= \text{vec}(\mathbf{W})^\top\left((\boldsymbol{\Omega}^{-\frac{1}{2}} \otimes \mathbf{X})^\top(\boldsymbol{\Omega}^{-\frac{1}{2}} \otimes \mathbf{X}) + \lambda\mathbf{I}_K \otimes \mathbf{I}_D\right.\\
&\quad \left. + \lambda_1\left(\boldsymbol{\Sigma}_c^{-\frac{1}{2}} \otimes \boldsymbol{\Sigma}_r^{-\frac{1}{2}}\right)^\top\left(\boldsymbol{\Sigma}_c^{-\frac{1}{2}} \otimes \boldsymbol{\Sigma}_r^{-\frac{1}{2}}\right)\right)\text{vec}(\mathbf{W})\\
&\quad - 2\text{vec}(\mathbf{W})^\top\left(\boldsymbol{\Omega}^{-\frac{1}{2}} \otimes \mathbf{X}^\top\right)\text{vec}\left[\left(\mathbf{H} - \mathbf{1b}^\top\right)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]\\
&\quad + \text{vec}\left[\left(\mathbf{H} - \mathbf{1b}^\top\right)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]^\top\text{vec}\left[(\mathbf{H} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]\\
&= \text{vec}(\mathbf{W})^\top(\boldsymbol{\Omega}^{-1} \otimes \mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{KD} + \lambda_1(\boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r))\text{vec}(\mathbf{W})\\
&\quad - 2\text{vec}(\mathbf{W})^\top\left(\boldsymbol{\Omega}^{-\frac{1}{2}} \otimes \mathbf{X}^\top\right)\text{vec}\left[(\mathbf{H} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]\\
&\quad + \text{vec}\left[(\mathbf{H} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]^\top\text{vec}\left[(\mathbf{H} - \mathbf{1b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\right]
\end{aligned}$$

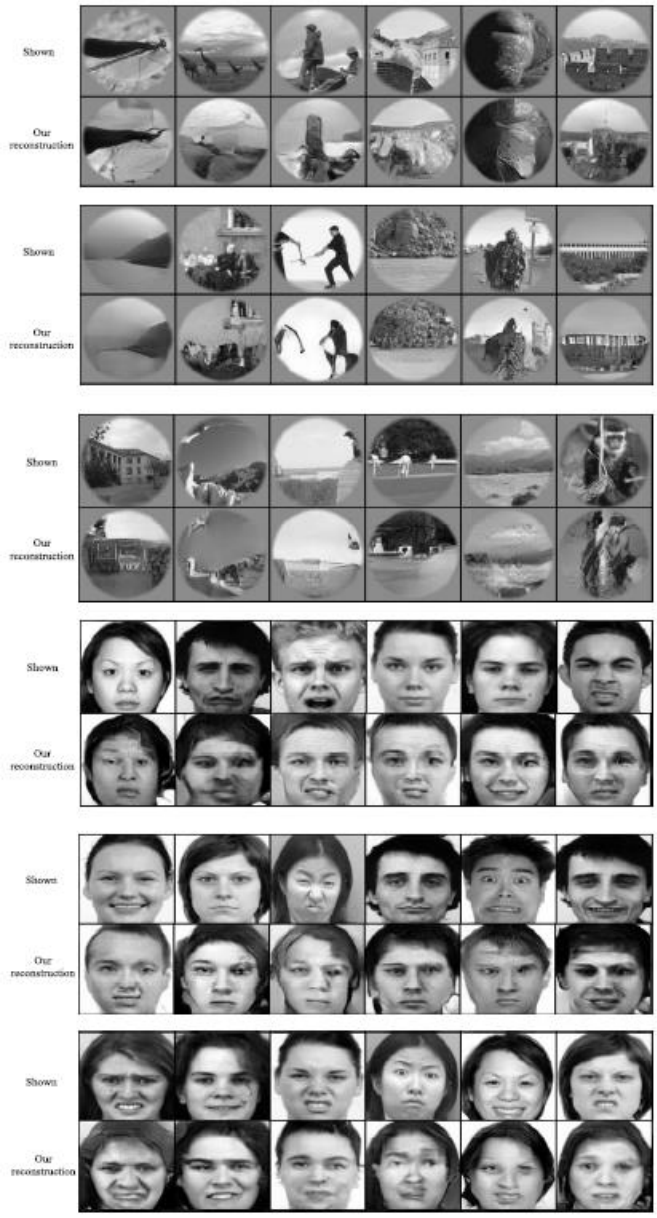

Fig. 15. Additional examples of reconstructed 128 × 128 natural images and human faces on the Vim-1 and FaceBold data sets, respectively.

$$
\begin{aligned}
&= \mathrm{vec}(\mathbf{W})^\top \underbrace{(\boldsymbol{\Omega}^{-1} \otimes \mathbf{X}^\top\mathbf{X} + \lambda \mathbf{I}_{KD} + \lambda_1(\boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r))}_{\mathbf{U}} \mathrm{vec}(\mathbf{W}) \\
&\quad - 2\mathrm{vec}(\mathbf{W})^\top \underbrace{\mathrm{vec}[\mathbf{X}^\top(\mathbf{H} - \mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1}]}_{\mathbf{V}} \\
&\quad + \mathrm{vec}\big[(\mathbf{H} - \mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\big]^\top \mathrm{vec}\big[(\mathbf{H} - \mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-\frac{1}{2}}\big].
\end{aligned}
$$

We can find that $\mathcal{L}_{\mathbf{W}}$ is a quadratic function of $\mathrm{vec}(\mathbf{W})$, hence the optimal $\hat{\mathbf{W}}$ is

$$
\begin{aligned}
\mathrm{vec}(\hat{\mathbf{W}}) &= \mathbf{U}^{-1}\mathbf{V} \\
&= \big[\boldsymbol{\Omega}^{-1} \otimes (\mathbf{X}^\top\mathbf{X}) + \lambda \mathbf{I}_{KD} + \lambda_1 \boldsymbol{\Sigma}_c^{-1} \otimes \boldsymbol{\Sigma}_r^{-1}\big]^{-1} \\
&\quad \times \mathrm{vec}(\mathbf{X}^\top(\mathbf{H} - \mathbf{1}\mathbf{b}^\top)\boldsymbol{\Omega}^{-1}).
\end{aligned}
$$

We can reshape $\mathrm{vec}(\hat{\mathbf{W}})$ into a $D \times K$ matrix to obtain $\hat{\mathbf{W}}$.

Appendix B
Additional Reconstruction Results

Additional reconstruction results on the Vim-1 and FaceBold data sets are given in Fig. 15.

Appendix C
More Comparisons on Each Individual Subject

See Tables VI and VII.

References

[1] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, Jul. 2017.

[2] H. Wen, J. Shi, Y. Zhang, K.-H. Lu, J. Cao, and Z. Liu, "Neural encoding and decoding with deep learning for dynamic natural vision," *Cerebral Cortex*, vol. 28, no. 12, pp. 4136–4160, 2017.

[3] G. Shen, T. Horikawa, K. Majima, and Y. Kamitani, "Deep image reconstruction from human brain activity," *PLOS Comput. Biol.*, vol. 15, no. 1, Jan. 2019, Art. no. e1006633.

[4] Y. Güçlütürk, U. Güçlü, K. Seeliger, S. Bosch, R. van Lier, and M. A. van Gerven, "Reconstructing perceived faces from brain activations with deep adversarial neural decoding," in *Proc. NIPS*, 2017, pp. 4246–4257.

[5] C. Du, C. Du, L. Huang, and H. He, "Reconstructing perceived images from human brain activities with Bayesian deep multiview learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2310–2323, Aug. 2019.

[6] T. Horikawa and Y. Kamitani, "Generic decoding of seen and imagined objects using hierarchical visual features," *Nature Commun.*, vol. 8, no. 1, p. 15037, Aug. 2017.

[7] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. CVPR*, Jun. 2015, pp. 5188–5196.

[8] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Proc. NIPS*, 2016, pp. 3387–3395.

[9] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. CVPR*, Jun. 2016, pp. 4829–4837.

[10] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. NIPS*, 2015, pp. 3483–3491.

[11] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[12] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. NIPS*, 2016, pp. 658–666.

[13] H. Huang, R. He, Z. Sun, and T. Tan, "IntroVAE: Introspective variational autoencoders for photographic image synthesis," in *Proc. NIPS*, 2018, pp. 52–63.

[14] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm

[15] J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, no. 5539, pp. 2425–2430, Sep. 2001.

[16] Y. Kamitani and F. Tong, "Decoding the visual and subjective contents of the human brain," *Nature Neurosci.*, vol. 8, no. 5, p. 679, 2005.

[17] M. A. J. van Gerven, B. Cseke, F. P. de Lange, and T. Heskes, "Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior," *NeuroImage*, vol. 50, no. 1, pp. 150–161, Mar. 2010.

[18] S. R. Damarla and M. A. Just, "Decoding the representation of numerical values from brain activation patterns," *Hum. Brain Mapping*, vol. 34, no. 10, pp. 2624–2634, Oct. 2013.

[19] E. Yargholi and G.-A. Hossein-Zadeh, "Brain decoding-classification of hand written digits from fMRI data employing Bayesian networks," *Frontiers Hum. Neurosci.*, vol. 10, no. 13, pp. 351–364, Jul. 2016.

[20] K. N. Kay, T. Naselaris, R. J. Prenger, and J. L. Gallant, "Identifying natural images from human brain activity," *Nature*, vol. 452, no. 7185, p. 352, 2008.

[21] Y. Miyawaki *et al.*, "Visual image reconstruction from human brain activity using a combination of multiscale local image decoders," *Neuron*, vol. 60, no. 5, pp. 915–929, Dec. 2008.

[22] T. Naselaris, R. J. Prenger, K. N. Kay, M. Oliver, and J. L. Gallant, “Bayesian reconstruction of natural images from Human brain activity,” *Neuron*, vol. 63, no. 6, pp. 902–915, Sep. 2009.

[23] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant, “Reconstructing visual experiences from brain activity evoked by natural movies,” *Current Biol.*, vol. 21, no. 19, pp. 1641–1646, Oct. 2011.

[24] A. J. Rothman, E. Levina, and J. Zhu, “Sparse multivariate regression with covariance estimation,” *J. Comput. Graph. Statist.*, vol. 19, no. 4, pp. 947–962, 2010.

[25] P. Rai, A. Kumar, and H. Daume, “Simultaneously leveraging output and task structures for multiple-output regression,” in *Proc. NIPS*, 2012, pp. 3185–3193.

[26] K.-A. Sohn and S. Kim, “Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization,” in *Proc. AISTATS*, 2012, pp. 1081–1089.

[27] H. Zhao, O. Stretcu, R. Negrinho, A. Smola, and G. Gordon, “Efficient multi-task feature and relationship learning,” in *Proc. NIPS*, 2017, pp. 777–787.

[28] A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in *Proc. NIPS*, vol. 2007, pp. 41–48.

[29] J. Chen, L. Tang, J. Liu, and J. Ye, “A convex formulation for learning shared structures from multiple tasks,” in *Proc. ICML*, 2009, pp. 137–144.

[30] P. Rai and H. Daumé, III, “Infinite predictor subspace models for multitask learning,” in *Proc. AISTATS*, 2010, pp. 613–620.

[31] A. Argyriou, M. Pontil, Y. Ying, and C. A. Micchelli, “A spectral regularization framework for multi-task structure learning,” in *Proc. NIPS*, 2008, pp. 25–32.

[32] A. Agarwal, S. Gerber, and H. Daume, “Learning multiple tasks using manifold regularization,” in *Proc. NIPS*, 2010, pp. 46–54.

[33] J. Chen, J. Zhou, and J. Ye, “Integrating low-rank and group-sparse structures for robust multi-task learning,” in *Proc. SIGKDD*, 2011, pp. 42–50.

[34] U. Güçlü and M. A. J. van Gerven, “Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream,” *J. Neurosci.*, vol. 35, no. 27, pp. 10005–10014, Jul. 2015.

[35] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, “Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence,” *Sci. Rep.*, vol. 6, no. 1, pp. 27755–27768, Sep. 2016.

[36] M. Eickenberg, A. Gramfort, G. Varoquaux, and B. Thirion, “Seeing it all: Convolutional network layers map the function of the human visual system,” *NeuroImage*, vol. 152, pp. 184–194, May 2017.

[37] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[38] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. NIPS*, 2014, pp. 2672–2680.

[39] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proc. ICML*, 2017, pp. 214–223.

[40] C. Du, C. Du, and H. He, “Sharing deep generative representation for perceived image reconstruction from human brain activity,” in *Proc. IJCNN*, May 2017, pp. 1049–1056.

[41] K. Han *et al.*, “Variational autoencoder: An unsupervised model for encoding and decoding fMRI activity in visual cortex,” *NeuroImage*, vol. 198, pp. 125–136, Sep. 2019.

[42] K. Seeliger, U. Güçlü, L. Ambrogioni, Y. Güçlütürk, and M. van Gerven, “Generative adversarial networks for reconstructing natural images from brain activity,” *NeuroImage*, vol. 181, pp. 775–785, Nov. 2018.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. NIPS*, 2012, pp. 1097–1105.

[44] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. Boca Raton, FL, USA: CRC Press, 2018.

[45] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, Jul. 2008.

[46] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in *Proc. ICML*, 2016, pp. 1558–1566.

[47] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, “CVAE-GAN: Fine-grained image generation through asymmetric training,” in *Proc. ICCV*, Oct. 2017, pp. 2745–2754.

[48] R. Zhang, S. Fattahi, and S. Sojoudi, “Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion,” in *Proc. ICML*, 2018, pp. 5766–5775.

[49] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014. [Online]. Available: https://openreview.net/forum?id=33X9fd2-9FyZd

[50] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, “Presentation and validation of the radboud faces database,” *Cognition Emotion*, vol. 24, no. 8, pp. 1377–1388, Dec. 2010.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[52] P. Chrabaszcz, I. Loshchilov, and F. Hutter, “A downsampled variant of ImageNet as an alternative to the CIFAR datasets,” 2017, *arXiv:1707.08819*. [Online]. Available: http://arxiv.org/abs/1707.08819

[53] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proc. ICCV*, Dec. 2015, pp. 3730–3738.

[54] Y. Fujiwara, Y. Miyawaki, and Y. Kamitani, “Modular encoding and decoding models derived from Bayesian canonical correlation analysis,” *Neural Comput.*, vol. 25, no. 4, pp. 979–1005, Apr. 2013.

[55] R. VanRullen and L. Reddy, “Reconstructing faces from fMRI patterns using deep generative neural networks,” *Commun. Biol.*, vol. 2, no. 1, pp. 1–10, 2019.

[56] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[58] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, Jun. 2016, pp. 770–778.

Changde Du received the B.E. degree in automatic control from the Beijing Information Science and Technology University, Beijing, China, in 2013, the M.S. degree in data mining from the University of Chinese Academy of Sciences, Beijing, in 2016, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, in 2019.

He has published over 20 peer-reviewed research articles in prestigious conferences and journals. His current research interests include machine learning, brain-inspired intelligence, computational neuroscience, and applications in computer vision.

Dr. Du received the Third Prize of Final Contest of National Collegiate Contest and International Invitational Tournament for Brain-inspired Computing and Application Award in 2017, the National Scholarship for Doctoral Students in 2018, the President Prize of Chinese Academy of Sciences for Excellent Ph.D. Graduates in 2019, and the Outstanding Doctoral Graduate of Beijing in 2019.

Changying Du received the B.Sc. degree from Central South University, Changsha, China, in 2008 and the Ph.D. degree in machine learning from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China, in 2015.

He is currently an Associate Professor with the Institute of Software, CAS. He was a Visiting Scholar with Purdue University, West Lafayette, IN, USA, from 2013 to 2014. He has published over 20 peer-reviewed research articles in prestigous conferences and journals.

Dr. Du serves as a program committee (PC) Member/Reviewer for Neural Information Processing Systems (NeurIPS) 2020, International Joint Conference on Artificial Intelligence (IJCAI) 2020, Association for the Advancement of Artificial Intelligence (AAAI) 2020, and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, etc.

Lijie Huang received the B.Sc. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, in 2010 and the Ph.D. degree in cognitive neuroscience from the State Key Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, Beijing, China, in 2016.

He is currently an Assistant Professor with the Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing. He works on science and technology challenges at the intersection of neuroscience, machine learning, and large-scale data analysis, e.g., how to read our mind using state-of-the-art machine learning techniques.

Haibao Wang received the B.Sc. degree in mathematics and applied mathematics from Anhui University, Anhui, China, in 2017, and the M.S. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2020.

His research interests include machine learning and computational neuroscience.

Dr. Wang received the National Scholarship for Undergraduates in 2016.

Huiguang He (Senior Member, IEEE) received the B.S. and M.S. degrees from Dalian Maritime University (DMU), Dalian, China, in 1994 and 1997, respectively, and the Ph.D. degree *(Hons.)* in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China.

He is currently a Full Professor with CASIA. He was an Associate Lecturer with DMU from 1997 to 1999, and a Postdoctoral Researcher with the University of Rochester, Rochester, NY, USA, from 2003 to 2004. He was a Visiting Professor with the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, from 2014 to 2015. He has published more than 150 peer-reviewed articles. His research interests include pattern recognition, medical image processing, and brain computer interface (BCI). His research has been supported by several research grants from the National Science Foundation of China.

Dr. He received the Excellent Ph.D. Dissertation of CAS in 2004, the National Science and Technology Award in 2003 and 2004, the Beijing Science and Technology Award in 2002 and 2003, the K. C. Wong Education Prizes in 2007 and 2009, the Jia-Xi Lu Young Talent Prize in 2009, and the Excellent Member of Youth Innovation Promotion Association, CAS, in 2016.