Article:
Zhou, Q, Du, C, Li, D et al. (3 more authors) (2022) Neural Encoding and Decoding with a Flow-based Invertible Generative Model. IEEE Transactions on Cognitive and Developmental Systems. ISSN 2379-8920

https://doi.org/10.1109/tcds.2022.3176977

Neural Encoding and Decoding with a Flow-based Invertible Generative Model

Qiongyi Zhou, Changde Du, Dan Li, Haibao Wang, Jian K. Liu, Huiguang He*, *Senior Member, IEEE*

***Abstract*—Recent studies on visual neural encoding and decoding have made significant progress, benefiting from the latest advances in deep neural networks having powerful representations. However, two challenges remain. First, the current decoding algorithms based on deep generative models always struggle with information losses, which may cause blurry reconstruction. Second, most studies model the neural encoding and decoding processes separately, neglecting the inherent dual relationship between the two tasks. In this paper, we propose a novel neural encoding and decoding method with a two-stage flow-based invertible generative model to tackle the above issues. First, a convolutional auto-encoder is trained to bridge the stimuli space and the feature space. Second, an adversarial cross-modal normalizing flow is trained to build up a bijective transformation between image features and neural signals, with local and global constraints imposed on the latent space to render cross-modal alignment. The method eventually achieves bi-directional generation of visual stimuli and neural responses with a combination of the flow-based generator and the auto-encoder. The flow-based invertible generative model can minimize information losses and unify neural encoding and decoding into a single framework. Experimental results on different neural signals containing spike signals and functional magnetic resonance imaging demonstrate that our model achieves the best comprehensive performance among the comparison models.**

***Index Terms*—Neural encoding, neural decoding, normalizing flow, cross-modal generation**

I. Introduction

RECENTLY, visual neural encoding and decoding have become increasingly important. Visual neural encoding refers to predicting neural responses to visual stimuli [1], [2], [3], while visual neural decoding means decoding the information of visual stimuli by identification [4], [5], classification [6], or reconstruction [7] [8]. Exploration of the intrinsic relationship between visual stimuli and neural representations can provide not only deep insight into the neural mechanisms and brain-inspired intelligence but also algorithmic support for practical applications, such as the brain-computer interface [9], [8] and the decoding of mental imagery [10].

Many studies have investigated the visual neural encoding and decoding of various types of neural recordings, mainly including the spike signals of retinal ganglion cells (RGCs) and functional magnetic resonance imaging (fMRI) of the brain visual cortex. Initially, the studies utilized the Linear-Nonlinear Poisson model (LNP) [11] and the Generalized Linear Model (GLM) [12] to model RGCs. Recently, encoding models based on machine learning [13], [14] and deep learning methods [1], [2] have unlocked advances in the RGC encoding of natural stimuli. The encoding models of the cerebral visual cortex typically consist of two stages [4]. First, the model extracts various feature maps from stimuli, such as Gabor wavelets [4] and features extracted by deep neural networks (DNNs) [3], [15], [16]. Then, linear mappings establish a link from the feature space to the activity space. Stimuli reconstruction is the most challenging task of visual neural decoding. As a result of the powerful capability of deep generative networks, visual reconstruction from neural spikes of RGCs [8] and voxel responses of the human visual cortex [7], [10] has made progress.

In general, deep learning has boosted research on visual neural encoding and decoding. However, two challenges remain.

1) Existing models based on Variational Auto-Encoders (VAEs) [17] and Generative Adversarial Networks (GANs) [18] suffer from information losses, which are essentially attributed to irreversible networks. One of the consequences is the blurry decoded images of the VAE-based models [7], [19]. Several studies [20], [21] have designed generators by combining VAE and GAN to avoid blurry reconstruction, but information losses will always occur as long as generative networks are irreversible.
2) Most algorithms are designed exclusively for encoding or decoding tasks, neglecting the dual relationship between them. Simultaneous training of dual tasks can be advantageous for cross-modal feature alignment and can also reduce training expenses and improve generalization, as reported by the study on multi-task learning [22]. In practice, the unified framework can favor the performance evaluation of neural encoding algorithms or visual neuroprostheses via visualization [8]. Studies in [15], [19] linked these dual tasks by developing encoding and decoding models that share feature space,

This work was supported in part by the National Natural Science Foundation of China (61976209, 61906188), CAS International Collaboration Key Project (173211KYSB20190024), and Strategic Priority Research Program of CAS (XDB32040000). (Corresponding author: Huiguang He)

Q. Zhou, C. Du, D. Li, and H. Wang are 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: zhouqiongyi2018@ia.ac.cn, changde.du@ia.ac.cn, danliai@hotmail.com, haibaow@hotmail.com).

J. K. Liu is with the School of Computing, University of Leeds, Leeds LS2 9JT, U.K. (e-mail: j.liu9@leeds.ac.uk).

H. 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, Shanghai 200031, China (e-mail: huiguang.he@ia.ac.cn).

but the two tasks were learned independently. A previous study in [23] simultaneously trained neural encoding and decoding using a Joint Multi-modal Variational Auto-Encoder (JMVAE) [24]. However, as mentioned in the first item, the main disadvantage of the model is the blurry reconstruction.

Almost all studies on visual neural encoding and decoding meet at least one of the challenges mentioned above. In this paper, we try remedies for both of the above issues.

1) We develop invertible network generators for neural encoding and decoding tasks. Specifically, we introduce alternative generative networks called normalizing flows [25] that are composed of a cascade of bijective transformations. Unlike VAE and GAN, normalizing flows ensure information integrity during generation. Additionally, normalizing flows can directly fit the real distribution through the Maximizing Likelihood Estimation (MLE). The approach is superior to the optimization of VAE on the lower bound to the marginal likelihood and the adversarial learning of GAN, which is at risk of unstable training.
2) We design a bi-directional cross-modal generation model to integrate neural encoding and decoding into a single framework and train them simultaneously. We regard neural encoding and decoding as processes of cross-modal generation. Neural encoding refers to the process of generating neural responses from stimuli, while neural decoding refers to the inverse process. Furthermore, the dual tasks share a common feature space.

Specifically, we propose a FLow-based Invertible Generative model (FLIG) for neural encoding and decoding. The model has two stages. In the first stage, a Convolutional Auto-Encoder (CAE) is trained to extract features from visual stimuli. In the second stage, a cross-modal normalizing flow with discriminators is trained to enable the invertible generation of image features and neural signals. The flow-based generator and the pretrained CAE are combined to establish bi-directional mappings between the stimulus space and the neural activity space. To close the gap between the heterogeneous modalities, we impose both local and global constraints on the latent variables of the cross-modal normalizing flow. The former is called the Mean Square Error (MSE), which minimizes the point-to-point distance between each pair of latent variables. The latter is a second-order Representational Similarity Analysis (RSA) [26], which forces the inter-modal representational similarity matrix to be similar to the intra-modal matrix. In this way, the latent representations of two modalities can be better aligned with each other. Cycle-consistency loss [27] is a common issue in cross-modal generation. In this paper, we exploit invertible normalizing flows as the generator. Thus, the bi-directional generation of neural signals and image features is completely cycle-consistent [28]. As long as the CAE is properly trained in the first stage, the cycle-consistency loss of the cross-modal generation of neural signals and stimuli can be negligible.

In summary, the main contributions of our paper are listed as follows.

1) We propose a novel neural encoding and decoding method with a flow-based invertible generative model that simultaneously learns two dual tasks with one training.
2) The flow-based model not only has a powerful capability to extract image features but also achieves bi-directional invertible generation while preserving details and cycle consistency.
3) We design local and global constraints on the modal-specific latent spaces to ensure domain alignment between the two heterogeneous modalities.
4) Experimental results on different neural signals demonstrate the powerful generalization of our model. FLIG achieves the best comprehensive performance among all comparison methods when various evaluation metrics are considered.

II. Related Work

A. Visual Neural Encoding and Decoding

Visual neural encoding and decoding of RGCs and the visual cortex have been of great importance.

1) Visual neural encoding: Classical methods of RGC neural encoding include linear-nonlinear models (LNs) and generalized linear models (GLMs). A linear filter and a nonlinear transformation are used to fit image stimuli into spike rates. Due to the complexity gap between models and real neural circuits, traditional models only fit responses to simple artificial stimuli well but show an unsatisfactory generalization to natural stimuli. Studies have attempted to use findings of neuroscience as prior knowledge [29] or resorted to machine learning methods [14] to strengthen the modeling capabilities. Recently, deep neural networks equipped with a powerful capacity for representation have emerged as a brand-new modeling technique for RGCs. Encoding models of RGCs based on convolutional neural networks (CNNs) [1], [30], [31] and recurrent neural networks (RNNs) [2] perform significantly better at describing RGC responses to natural stimuli. The neural encoding of the visual cortex is more complex. Classical models contain two stages [4]. First, the stimuli are projected to the feature space via a nonlinear transformation. The feature maps can be Gabor wavelets [4], semantic features [5], or DNN-based features [3]. In the second stage, the image features are regressed onto the voxel responses through linear mappings. The state-of-the-art method is called the feature-weighted Receptive Field (fwRF) [3]. It computes a weighted sum of image features within a 2D Gaussian receptive field that is estimated for each voxel and regresses it to the corresponding voxel response. Note that the voxel-wise encoding pattern will incur considerable training expenses.

2) Visual neural decoding: The conventional method for visual neural decoding is called pixel-wise nonlinear regression [32], [33], which can only handle simple artificial stimuli but does not work well on natural images. In computer vision, deep generative networks, such as VAE and GAN, have demonstrated enormous potential for generating vivid images. Thus, many studies on neural decoding establish a connection

between the latent representations of generative models and neural activity and utilize the generators to reconstruct images [7], [10], [19], [20], [23], [34]. However, the reconstruction of VAE-based models tends to be blurry, and GAN-based models usually suffer from unstable training.

Unlike previous studies based on irreversible networks, this paper utilizes flow-based invertible networks to minimize information losses and cycle-consistency losses during neural encoding and decoding. In addition, until now, research on neural encoding and decoding has been segregated, neglecting their dual relationship. In this paper, we unify neural encoding and decoding into one framework with the bi-directional generation model and simultaneously learn the dual tasks. Our work is related to the CDDG method in [23]. However, the fundamental distinction between FLIG and CDDG is that CDDG builds up a neural encoding and decoding model with irreversible VAE, whereas FLIG in this paper is based on flow-based invertible networks.

B. Cross-Modal Generation

Many algorithms for multi-view learning have been proposed to achieve cross-modal generation. The Deep Canonically Correlated Auto-Encoder (DCCAE) proposed in [35] is the deep-learning version of canonical correlation analysis [36]. DCCAE learns a shared latent space from which each modality is reconstructed. Due to the correlation-based optimization, the latent representation only preserves the modal-shared information and abandons the modal-specific information. Such a design is inappropriate for cross-modal generation. [24] proposed JMVAEs to perform bi-directional cross-modal generation. Unlike DCCAE, JMVAE learns not only two modal-specific latent spaces but also a modal-shared latent space. Then, the Kullback-Leibler divergence is applied to make the distributions aligned. [27] came up with a GAN using cycle-consistent loss (CycleGAN) to reduce the solution space and enable unpaired image-to-image translation. Furthermore, AlignFlow proposed in [28] substitutes the generators of CycleGAN with normalizing flows. Due to the invertibility of the normalizing flows, cycle consistency losses of the generation no longer exist.

In comparison to CycleGAN and AlignFlow, our work emphasizes diminishing the heterogeneity between modalities, which is a negligible issue for the image-to-image translation task but an inevitable problem for the neural encoding and decoding tasks. Therefore, we introduce local and global constraints on latent space to shrink the modal gap. In addition, the CAE feature extractor can reduce the image dimension to that of the neural signals before feeding the images into the normalizing flows. The design adapts the model to the cross-modal generation of two different-dimensional modalities that AlignFlow cannot handle.

III. Method

A. Overview

In this paper, we study visual neural encoding and decoding problems within a single model. For a neural dataset, visual stimuli and neural signals are represented as $\mathbf{x} \in \mathbb{R}^{N\times P\times P}$ and $\mathbf{s} \in \mathbb{R}^{N\times M}$, respectively. Note that N, P and M are the sample size, the image resolution and the dimension of neural signals. Paired samples of two modalities build up an i.i.d. multi-modal dataset $(\mathbf{x}, \mathbf{s}) = \{(\mathbf{x}_i, \mathbf{s}_i)\}_{i=1}^{N}$. Our study aims to learn a bi-directional mapping between visual stimuli and neural activity, such that the model can not only generate the predicted neural activity $\hat{\mathbf{s}}_i$ from $\mathbf{x}_i$ via the forward mapping but also produce the reconstructed stimulus $\hat{\mathbf{x}}_i$ from $\mathbf{s}_i$ via the backward mapping.

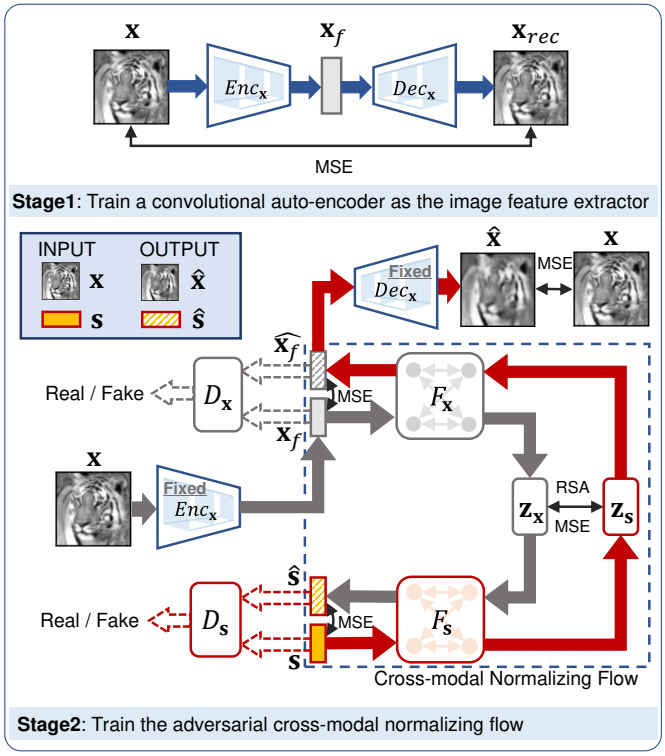

Fig. 1: The diagram of the two-stage training of our FLIG model (best viewed in color). The top figure shows the training of a convolutional auto-encoder that serves as the image feature extractor in the first stage. The bottom figure shows the training of the adversarial cross-modal normalizing flow in the second stage. $F_{\mathbf{x}}, F_{\mathbf{s}}$ and $D_{\mathbf{x}}, D_{\mathbf{s}}$ are the flow-based generators and discriminators of the image domain and the neural signal domain, respectively. $Enc_{\mathbf{x}}, Dec_{\mathbf{x}}$ with fixed parameters are the encoder and decoder trained in the first stage. $\mathbf{z}_{\mathbf{x}}$ and $\mathbf{z}_{\mathbf{s}}$ are the latent variables. The solid arrows in gray and red denote the routes of neural encoding and neural decoding, respectively. The dotted arrows in gray and red denote the discrimination of the image domain and the neural signal domain. The black bi-directional arrows in both figures indicate the constraints imposed on the variables pointed by the arrows, with annotations indicating the type of constraints.

Here, we propose a two-stage flow-based invertible generative model to achieve the goal. The model diagram is shown in Fig. 1. In the first stage, a convolutional auto-encoder containing the encoder $Enc_{\mathbf{x}}$ and the decoder $Dec_{\mathbf{x}}$ is trained to extract features $\mathbf{x}_f$ from the visual stimuli $\mathbf{x}$. Then, the parameters of $Enc_{\mathbf{x}}$ and $Dec_{\mathbf{x}}$ are fixed. In the second stage, an adversarial cross-modal normalizing flow composed of two

TABLE I: Definition of frequently used symbols.

Symbol	Definition
N	Sample size
P	Image resolution
M	Dimension of neural signals
$\mathbf{x}$	Visual stimuli
$\mathbf{s}$	Neural signals
$\hat{\mathbf{x}}$	Images generated by $\mathbf{s}$
$\hat{\mathbf{s}}$	Neural signals generated by $\mathbf{x}$
$\mathbf{x}_{rec}$	Images reconstructed from $\mathbf{x}$
$\mathbf{x}_f$	Image features generated by $\mathbf{x}$
$\hat{\mathbf{x}}_f$	Image features generated by $\mathbf{s}$
$\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{s}$	Latent variable of $\mathbf{x}, \mathbf{s}$
$\mathbf{z}_{\mathbf{x}_i}$	Latent variable of i-th visual stimuli
$\mathbf{z}_{\mathbf{s}_i}$	Latent variable of i-th neural signals
$K_\mathbf{x}, K_\mathbf{s}$	The number of layers of networks $F_\mathbf{x}, F_\mathbf{s}$
$\mathbf{z}_0, \cdots, \mathbf{z}_{K_\mathbf{x}}$	Input of each layers of $F_\mathbf{x}$
$\mathbf{z}_0, \cdots, \mathbf{z}_{K_\mathbf{s}}$	Input of each layers of $F_\mathbf{s}$
$F_\mathbf{x}$	The normalizing flow of visual stimuli
$F_\mathbf{s}$	The normalizing flow of neural signals
$D_\mathbf{x}$	The discriminator of visual stimuli
$D_\mathbf{s}$	The discriminator of neural signals
$F_\mathbf{x}^{-1}$	The inverse transformation of $F_\mathbf{x}$
$F_\mathbf{s}^{-1}$	The inverse transformation of $F_\mathbf{s}$
$\theta^{F_\mathbf{x}}, \theta^{F_\mathbf{s}}, \theta^{D_\mathbf{x}}, \theta^{D_\mathbf{s}}$	Parameters of networks $F_\mathbf{x}, F_\mathbf{s}, D_\mathbf{x}, D_\mathbf{s}$
$Enc_\mathbf{x}, Dec_\mathbf{x}$	Encoder and Decoder of the CAE
ϕ, γ	Parameters of networks $Enc_\mathbf{x}, Dec_\mathbf{x}$
$\mathbf{M}_{\mathbf{xx}}$	Intra-modal representation dissimilarity matrix
$\mathbf{M}_{\mathbf{xs}}$	Inter-modal representation dissimilarity matrix
$f_{\mathbf{x}_1}(\cdot), \cdots, f_{\mathbf{x}_{K_\mathbf{x}}}(\cdot)$	Layers of network $F_\mathbf{x}$
$f_{\mathbf{s}_1}(\cdot), \cdots, f_{\mathbf{s}_{K_\mathbf{s}}}(\cdot)$	Layers of network $F_\mathbf{s}$
$G_{\mathbf{x}\to\mathbf{s}}$	$F_\mathbf{s} F_\mathbf{x}^{-1}$
$\kappa(G_{\mathbf{x}\to\mathbf{s}})$	Condition number of $G_{\mathbf{x}\to\mathbf{s}}$

normalizing flows, $F_\mathbf{x}$ and $F_\mathbf{s}$, and two discriminators, $D_\mathbf{x}$ and $D_\mathbf{s}$, are trained to achieve the invertible generation of image features $\mathbf{x}_f$ and the corresponding neural signals $\mathbf{s}$ along the red and gray routes in Fig. 1. The discriminators play an adversarial game with the flow-based generators to enhance the sample quality. To bridge the modality gap, we force MSE and RSA-based constraints on the latent variables, $\mathbf{z}_\mathbf{x}$ and $\mathbf{z}_\mathbf{s}$. Given the pairwise visual stimuli $\mathbf{x}$ and neural signals $\mathbf{s}$ as the input, the adversarial cross-modal normalizing flow, combined with the fixed $Enc_\mathbf{x}$ and $Dec_\mathbf{x}$, generates the corresponding reconstruction $\hat{\mathbf{s}}$ and $\hat{\mathbf{x}}$ along the gray and red routes in Fig. 1, respectively. We elaborate on the method in the following sections. We list the symbols that will be used frequently in the following with their definitions in Table I in advance.

B. Image feature extractor

Due to the domain gap, it is challenging to learn a direct mapping between visual stimuli and neural activity. Previous studies on neural encoding and decoding [7], [8], [19], [20] have demonstrated that an efficient solution is to project visual stimuli and neural activity onto the intermediate feature space. Thus, in the first stage of our model, an image feature extractor is designed to serve as the intermediate space between the stimuli space and the neural response space.

Here, we use a convolutional auto-encoder to extract image features $\mathbf{x}_f$. CAE plays a prominent role in perceiving high-level image representations in an unsupervised way. It is comprised of an encoder $Enc_\mathbf{x}$ and a decoder $Dec_\mathbf{x}$. The encoder reduces the dimension of the input data and the decoder recovers the data from the feature space. The convolution and dropout operations in the encoder are responsible for extracting low-dimensional features, while the upsampling and deconvolution operations in the decoder are used to reconstruct images from features. The reconstruction loss leads the model to learn a latent representation. The loss function is

$$\mathcal{L}_{\text{CAE}} = \|\mathbf{x} - \mathbf{x}_{rec}\|_2^2 \tag{1}$$

where $\mathbf{x}$ and $\mathbf{x}_{\text{rec}}$ are the original and reconstructed images, respectively. Note that the dimension of the extracted features is reduced to M since the invertible networks that bridge the feature domain and the neural activity domain in the second stage do not permit dimension changes.

C. Normalizing flows

In the second step of our model, we build a mapping between the feature space and the neural response space. Unlike VAE and GAN, normalizing flows are composed of invertible networks and can directly estimate the real data distribution. Therefore, normalizing flows are chosen to work as the bridge between the feature domain and the neural response domain.

Normalizing flows can transform a simple probability density distribution into a highly complex distribution through a series of invertible mapping networks. Given a random variable $\mathbf{z}_0$, an invertible mapping $f_1 : \mathbb{R}^D \to \mathbb{R}^D$ projects $\mathbf{z}_0$ into $\mathbf{z}_1$. According to the change of variable formula, the relationship between the distribution $p(\mathbf{z}_0)$ and $p(\mathbf{z}_1)$ is

$$p(\mathbf{z}_1) = p(\mathbf{z}_0)\left|\det\frac{\partial f_1^{-1}}{\partial \mathbf{z}_1}\right| = p(\mathbf{z}_0)\left|\det\frac{\partial f_1}{\partial \mathbf{z}_0^T}\right|^{-1}. \tag{2}$$

A concatenation of K invertible networks $f_1, f_2, \cdots, f_K$ can enhance the expression ability of flows. Then, the distribution of the target variable $\mathbf{z}_K = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0)$ is

$$p(\mathbf{z}_K) = p(\mathbf{z}_0)\prod_{i=1}^{K}\left|\det\frac{\partial f_i}{\partial \mathbf{z}_{i-1}^T}\right|^{-1}. \tag{3}$$

The invertible property of the network, on the one hand, ensures information integrity during generation and, on the other hand, signifies that the dimensions of $\mathbf{z}_K$ and $\mathbf{z}_0$ must be equal. Normalizing flows build a bridge between the initial and the obtained distributions through the determinants of the Jacobian matrices. As generative models, normalizing flows can be used to generate data using MLE. In practice, the negative log-likelihood of the distribution of the generated data $p(\mathbf{z}_K)$ is minimized with the loss function

$$\mathcal{L}_{\text{MLE}} = -\ln p(\mathbf{z}_K) = -\ln p(\mathbf{z}_0) + \sum_{i=1}^{K}\left|\det\frac{\partial f_i}{\partial \mathbf{z}_{i-1}^T}\right|. \tag{4}$$

In this paper, we use the Real-valued Non-Volume Preserving (Real-NVP) transformations [37] to form normalizing flows because they can balance good model performance, low computational cost, and high convergence speed compared to other transformations [38], [39]. The transformation $f_i(\cdot)$ is

$$\begin{aligned}\mathbf{z}_i^{1:d} &= \mathbf{z}_{i-1}^{1:d},\\ \mathbf{z}_i^{d+1:D} &= \mathbf{z}_{i-1}^{d+1:D} \odot \exp(s(\mathbf{z}_{i-1}^{1:d})) + t(\mathbf{z}_{i-1}^{1:d})\end{aligned} \tag{5}$$

where $d < D$, and the vector $\mathbf{z}_i \in \mathbb{R}^D$ is sliced into two parts with the superscripts $(1 : d)$ and $(d + 1 : D)$. $\odot$ is an element-wise product, and $s(\cdot), t(\cdot)$ denote the scale and translation from $\mathbb{R}^d \to \mathbb{R}^{D-d}$. The transformation is an affine coupling layer since the second part of the vector is transformed depending on the first part. In addition, the first part that remains unchanged in this layer will be updated in the subsequent layer, while the second part that changes in this layer will remain unchanged in the subsequent layer. The triangular Jacobian matrix of the transformation is

$$\frac{\partial f_i}{\partial \mathbf{z}_{i-1}^T} = \begin{bmatrix} \mathbb{I}_d & \mathbf{0} \\ \frac{\partial \mathbf{z}_{i-1}^{d+1:D}}{\partial (\mathbf{z}_{i-1}^{1:d})^T} & \text{diag}(\exp[s(\mathbf{z}_{i-1}^{1:d})]) \end{bmatrix}, \tag{6}$$

whose determinant is easy to calculate.

D. Adversarial cross-modal normalizing flows

As shown in Eq. (3), single-modal normalizing flows can only generate data from latent variables. To achieve cross-modal generation of $\mathbf{x}_f$ and $\mathbf{s}$, we combine two modal-specific normalizing flows.

Specifically, the normalizing flow of the stimuli domain is $F_{\mathbf{x}} = f_{\mathbf{x}_{K_{\mathbf{x}}}} \circ \cdots \circ f_{\mathbf{x}_2} \circ f_{\mathbf{x}_1}(\cdot)$ with $K_{\mathbf{x}}$ layers, and the normalizing flow of the neural signal domain is $F_{\mathbf{s}} = f_{\mathbf{s}_{K_{\mathbf{s}}}} \circ \cdots \circ f_{\mathbf{s}_2} \circ f_{\mathbf{s}_1}(\cdot)$ with $K_{\mathbf{s}}$ layers. Two modalities share one initial distribution $p(\mathbf{z}_0)$. Thus, the generated neural responses are $\hat{\mathbf{s}} = F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\mathbf{x}_f))$, and the generated image features are $\widehat{\mathbf{x}_f} = F_{\mathbf{x}}(F_{\mathbf{s}}^{-1}(\mathbf{s}))$. The cross-modal normalizing flow can be trained by minimizing the loss function

$$\begin{aligned} \mathcal{L}_{\text{MLE}} =& \mathcal{L}_{\text{MLE}_{\mathbf{x}}} + \mathcal{L}_{\text{MLE}_{\mathbf{s}}} \\ =& -\ln p(\mathbf{x}_f) - \ln p(\mathbf{s}) \\ =& -\ln p(\mathbf{z}_0) + \sum_{i=1}^{K_{\mathbf{x}}} \left| \det \frac{\partial f_{\mathbf{x}_i}}{\partial \mathbf{z}_{i-1}^T} \right| \\ & -\ln p(\mathbf{z}_0) + \sum_{i=1}^{K_{\mathbf{s}}} \left| \det \frac{\partial f_{\mathbf{s}_i}}{\partial \mathbf{z}_{i-1}^T} \right|. \end{aligned} \tag{7}$$

Normalizing flows give an explicit optimization of the real data distribution, but the generation quality is not very outstanding [40]. In comparison, GAN only has an implicit estimation of the real data distribution but is good at generating high-fidelity samples. This observation motivates us to incorporate adversarial learning into the cross-modal normalizing flow. In our model, the flow-based generators $F_{\mathbf{x}}, F_{\mathbf{s}}$ play an adversarial game with two modal-specific discriminators $D_{\mathbf{x}}, D_{\mathbf{s}}$. The discriminators are responsible for distinguishing fake data from real data. For example, $D_{\mathbf{x}}$ is expected to label the image features of the stimuli as real data and label those generated from the neural responses as fake data. The generators attempt to generate fake data that the discriminators will tag as real. Suppose that $p_{\text{data}}(\mathbf{x}_f)$ and $p_{\text{data}}(\mathbf{s})$ are real data distributions of image features and neural responses, respectively. The loss function of the generators is

$$\begin{aligned} \mathcal{L}_F =& \mathcal{L}_{\text{MLE}_{\mathbf{x}}} + \mathcal{L}_{\text{MLE}_{\mathbf{s}}} + \mathcal{L}_{G_{\mathbf{x}}} + \mathcal{L}_{G_{\mathbf{s}}} \\ =& \mathcal{L}_{\text{MLE}_{\mathbf{x}}} + \mathcal{L}_{\text{MLE}_{\mathbf{s}}} \\ & + \mathbb{E}_{\mathbf{s} \sim p_{\text{data}}(\mathbf{s})}[\log(1 - D_{\mathbf{x}}(F_{\mathbf{x}}(F_{\mathbf{s}}^{-1}(\mathbf{s}))))] \\ & + \mathbb{E}_{\mathbf{x}_f \sim p_{\text{data}}(\mathbf{x}_f)}[\log(1 - D_{\mathbf{s}}(F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\mathbf{x}_f))))]. \end{aligned} \tag{8}$$

Although adversarial learning is introduced to the cross-modal normalizing flows, the training stability of the generators can still be guaranteed by the intrinsically stable flow-based optimization in the first two items in Eq. (8). The loss function of the discriminators is

$$\begin{aligned} \mathcal{L}_D =& \mathcal{L}_{D_{gp-\mathbf{x}}} + \mathcal{L}_{D_{gp-\mathbf{s}}} + \mathcal{L}_{D_{\mathbf{x}}} + \mathcal{L}_{D_{\mathbf{s}}} \\ =& \mathcal{L}_{D_{gp-\mathbf{x}}} + \mathcal{L}_{D_{gp-\mathbf{s}}} \\ & - \mathbb{E}_{\mathbf{x}_f \sim p_{\text{data}}(\mathbf{x}_f)}[\log(D_{\mathbf{x}}(\mathbf{x}_f))] \\ & - \mathbb{E}_{\mathbf{s} \sim p_{\text{data}}(\mathbf{s})}[\log(1 - D_{\mathbf{x}}(F_{\mathbf{x}}(F_{\mathbf{s}}^{-1}(\mathbf{s}))))] \\ & - \mathbb{E}_{\mathbf{s} \sim p_{\text{data}}(\mathbf{s})}[\log(D_{\mathbf{s}}(\mathbf{s}))] \\ & - \mathbb{E}_{\mathbf{x}_f \sim p_{\text{data}}(\mathbf{x}_f)}[\log(1 - D_{\mathbf{s}}(F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\mathbf{x}_f))))], \end{aligned} \tag{9}$$

where $\mathcal{L}_{D_{gp-\mathbf{x}}}$ and $\mathcal{L}_{D_{gp-\mathbf{s}}}$ are gradient penalties on the discriminators for a stable training [41]. See Appendix B for details.

E. Constraints on the inter-modal gap

Since two modalities share one initial distribution $p(\mathbf{z}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, the latent representations of two modalities, $\mathbf{z}_{\mathbf{x}} = F_{\mathbf{x}}^{-1}(\mathbf{x}_f)$ and $\mathbf{z}_{\mathbf{s}} = F_{\mathbf{s}}^{-1}(\mathbf{s})$, both obey the isotropic Gaussian distribution. Despite the identical distributions, $\mathbf{z}_{\mathbf{x}_i}, \mathbf{z}_{\mathbf{s}_i}$ of the i-th sample may not be adjacent due to modality heterogeneity. To assure that $\hat{\mathbf{s}}_i$, generated from $\mathbf{x}_{f_i}$, can recover $\mathbf{s}_i$ as well as possible and vice versa given any pair of samples $\langle \mathbf{x}_{f_i}, \mathbf{s}_i \rangle$, we need to impose additional constraints on the latent space of the cross-modal normalizing flow. Here, local and global constraints are used to close the inter-modal gap.

First, the Mean Squared Error (MSE) loss brings the Euclidean distance between $\mathbf{z}_{\mathbf{x}}$ and $\mathbf{z}_{\mathbf{s}}$ closer. The loss function is

$$\mathcal{L}_{\mathbf{z}} = \|\mathbf{z}_{\mathbf{x}} - \mathbf{z}_{\mathbf{s}}\|_2^2. \tag{10}$$

The MSE, as a local constraint, explicitly ensures the point-to-point alignment on the latent space between each pairwise $\langle \mathbf{x}_{f_i}, \mathbf{s}_i \rangle$.

Second, we take advantage of the Representation Similarity Analysis (RSA) to improve the correlation between $\mathbf{z}_{\mathbf{x}}$ and $\mathbf{z}_{\mathbf{s}}$. RSA was proposed in the field of neuroscience to quantitatively probe the relevance of brain-activity measurement, behavioral measurement, and computational modeling [26]. RSA is typically performed by Representation Dissimilarity Matrices (RDMs). Suppose $\rho_{\mathbf{x}\mathbf{s}_{ij}} = \mathbf{z}_{\mathbf{x}_i} \otimes \mathbf{z}_{\mathbf{s}_j}$. $\otimes$ denotes the computation of the cosine similarity and $\rho_{\mathbf{x}\mathbf{s}_{ij}} \in [-1, 1]$. The inter-modal RDM is $\mathbf{M}_{\mathbf{x}\mathbf{s}}[i, j] = (1 - \rho_{\mathbf{x}\mathbf{s}_{ij}})/2$. The diagonal elements should be close to zero since the pairwise latent variables should be as correlated as possible. The MSE mentioned above can have such an effect. However, global restrictions on latent spaces are in demand. Suppose that there

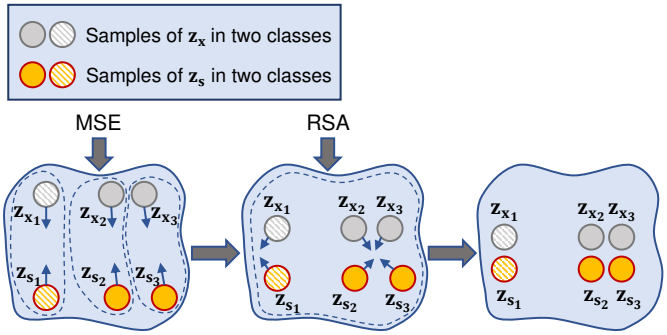

Fig. 2: A demonstration of the effects of the MSE and the second-order RSA on learning latent representations (better viewed in color). The dots in gray and yellow respectively denote three samples of $\mathbf{z_x}$ and $\mathbf{z_s}$. The pairwise samples have the same subscript numbers. The fill style is identical across samples in the same class. The arrows indicate the effects of the added constraints on each variable within the dotted circles.

are two similar stimuli $\mathbf{x}_i, \mathbf{x}_j$ with the same category. Most likely, the neural responses $\mathbf{s}_i, \mathbf{s}_j$ are similar. For a well-trained model, two pairs of latent variables, $\langle \mathbf{z}_{\mathbf{x}_i}, \mathbf{z}_{\mathbf{s}_j} \rangle$ and $\langle \mathbf{z}_{\mathbf{s}_i}, \mathbf{z}_{\mathbf{s}_j} \rangle$, are expected to be correlated. Therefore, the similarity between the observed variables could be exploited to guide the learning of the latent representations.

Here, we resort to the second-order RDM [26]. We introduce an intra-modal RDM $\mathbf{M_{xx}}$ in addition to the inter-modal RDM $\mathbf{M_{xs}}$. The elements are

$$\mathbf{M_{xx}}[i,j] = \frac{1-\rho_{\mathbf{xx}_{ij}}}{2} = \frac{1-\mathbf{z}_{\mathbf{x}_i} \otimes \mathbf{z}_{\mathbf{x}_j}}{2},$$
$$\mathbf{M_{xs}}[i,j] = \frac{1-\rho_{\mathbf{xs}_{ij}}}{2} = \frac{1-\mathbf{z}_{\mathbf{x}_i} \otimes \mathbf{z}_{\mathbf{s}_j}}{2}.$$

Then, the Frobenius norm of the difference matrix is minimized,

$$\mathcal{L}_{\text{RSA}} = \|\mathbf{M_{xx}} - \mathbf{M_{xs}}\|_{\text{F}}. \tag{11}$$

The more similar $\mathbf{x}_i$ and $\mathbf{x}_j$ are, the larger $\rho_{\mathbf{xx}_{ij}}$ is. According to Eq. (11), $\rho_{\mathbf{xs}_{ij}}$ will be larger. Consequently, $\mathbf{z}_{\mathbf{s}_i}$ and $\mathbf{z}_{\mathbf{s}_j}$ will be more similar. The opposite situation has the opposite inferences.

For the sake of clarity, Fig. 2 demonstrates the effects of MSE and RSA-based constraints on latent representation learning . It is assumed that there are three pairs of samples $\langle \mathbf{z}_{\mathbf{x}_1}, \mathbf{z}_{\mathbf{s}_1} \rangle, \langle \mathbf{z}_{\mathbf{x}_2}, \mathbf{z}_{\mathbf{s}_2} \rangle, \langle \mathbf{z}_{\mathbf{x}_3}, \mathbf{z}_{\mathbf{s}_3} \rangle$ and that the last two pairs belong to the same semantic category. MSE produces local effects that bring each pair of samples closer, while the second-order RSA has global effects on all samples. Specifically, the RSA-based constraint minimizes the intra-class distance, maximizes the inter-class distance, and finally makes the latent representation of two domains more consistent. These two kinds of constraints are complementary and indispensable.

F. Synchronous convergence

In addition to the constraints on the latent variables, constraints on the observed variables are imposed to ensure the high accuracy of neural encoding and decoding. Besides, the regularization technique, called Jacobian Clamping [28], [42], is utilized to guarantee synchronous convergence of neural encoding and decoding.

Constraints are imposed on $\mathbf{x}$ and $\hat{\mathbf{x}}$ of the image domain, $\mathbf{x}_f$ and $\widehat{\mathbf{x}_f}$ of the feature domain, and $\mathbf{s}$ and $\hat{\mathbf{s}}$ of the neural activity domain. Concretely, the following three losses are added to the holistic loss:

$$\mathcal{L}_{\mathbf{x}} = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2, \tag{12}$$
$$\mathcal{L}_{\mathbf{x}_f} = \|\widehat{\mathbf{x}_f} - \mathbf{x}_f\|_2^2, \tag{13}$$
$$\mathcal{L}_{\mathbf{s}} = \|\hat{\mathbf{s}} - \mathbf{s}\|_2^2, \tag{14}$$

separately minimizing the reconstruction error of images, image features and neural signals. The residuals in Eq. (13) and (14) are coupled to some extent. The derivation is in Appendix A. Let $G_{\mathbf{x}\to\mathbf{s}} = F_{\mathbf{s}} F_{\mathbf{x}}^{-1}$ and $\kappa(G_{\mathbf{x}\to\mathbf{s}})$ be the condition number of $G_{\mathbf{x}\to\mathbf{s}}$ [43]. We have

$$\frac{1}{\kappa(G_{\mathbf{x}\to\mathbf{s}})} \frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|} \le \frac{\|\mathbf{s}\|}{\|\hat{\mathbf{s}}\|} \le \kappa(G_{\mathbf{x}\to\mathbf{s}}) \frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|}. \tag{15}$$

See Appendix A for details of this coupling inequality. The inequality in Eq. (15) means that if $\kappa(G_{\mathbf{x}\to\mathbf{s}})$ is large, the fluctuation scale of $\|\mathbf{s}\|/\|\hat{\mathbf{s}}\|$ may be uncertain, although $\|\widehat{\mathbf{x}_f}\|/\|\mathbf{x}_f\| \to 1$. The specific phenomenon is that the reconstruction of one modality is already perfect, while that of the other modality remains unsatisfactory. As a result, the convergence of the two tasks might be out of sync. One solution is to control the condition number $\kappa(G_{\mathbf{x}\to\mathbf{s}})$ within a region, but it is unfeasible. A more plausible solution is to limit the fluctuation ratio of $\mathbf{s}$ and $\mathbf{x}_f$. The Jacobian Clamping regularization technique proposed by [42] can attain the objective. Although the Jacobian Clamping regularization was originally proposed to ensure a more stable training of GAN, we adopt it in our model for synchronous convergence of neural encoding and decoding. Specifically, infusing random noise $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into the input variable $\mathbf{x}_f$, i.e., $\mathbf{x}'_f = \mathbf{x}_f + \delta/\|\delta\|$, the ensuing fluctuation ratio of the output variables is

$$Q = \frac{\|G_{\mathbf{x}\to\mathbf{s}}(\mathbf{x}_f) - G_{\mathbf{x}\to\mathbf{s}}(\mathbf{x}'_f)\|}{\|\mathbf{x}_f - \mathbf{x}'_f\|}.$$

The regularization is implemented as

$$\mathcal{L}_{\text{JC}} = (\max(Q, \lambda^+) - \lambda^+)^2 + (\min(Q, \lambda^-) - \lambda^-)^2 \tag{16}$$

to control the change ratio Q to lie within $[\lambda^-, \lambda^+]$. We can adjust the interval to make $\|G_{\mathbf{x}\to\mathbf{s}}(\mathbf{x}_f) - G_{\mathbf{x}\to\mathbf{s}}(\mathbf{x}'_f)\|$ small when $\|\mathbf{x}_f - \mathbf{x}'_f\|$ is small. Therefore, the Jacobian Clamping can facilitate synchronous convergence of the multi-tasks learned by the cross-modal normalizing flows by restricting the residual ratio Q.

G. Optimizing Strategy

The optimization process contains two stages. Algorithm 1 outlines the entire training course. In the first stage, CAE is trained to extract image features using an encoder and a decoder. The loss function of the first stage is $\mathcal{L}_1 = \mathcal{L}_{\text{CAE}}$. In the second stage, the network parameters of CAE are fixed. The parameters of generators $F_{\mathbf{x}}, F_{\mathbf{s}}$, and the parameters of

Algorithm 1 FLow-based Invertible Generative networks

Input: Paired original variables (visual stimuli, neural signals)
$(\mathbf{x}, \mathbf{s}) = \{(\mathbf{x}_1, \mathbf{s}_1), (\mathbf{x}_2, \mathbf{s}_2), \cdots, (\mathbf{x}_N, \mathbf{s}_N)\}$
Output: Paired reconstructed variables
$(\hat{\mathbf{x}}, \hat{\mathbf{s}}) = \{(\hat{\mathbf{x}}_1, \hat{\mathbf{s}}_1), (\hat{\mathbf{x}}_2, \hat{\mathbf{s}}_2), \cdots, (\hat{\mathbf{x}}_N, \hat{\mathbf{s}}_N)\}$
S1: Training of the image feature extractor CAE
Initialize the parameters (ϕ, γ) of the networks $Enc_{\mathbf{x}}$ and $Dec_{\mathbf{x}}$.
Set latent dimension M, batch size B_1, learning rate μ_1, maximum iteration number T_1
for t_1 iteration **do**
  Compute $\mathbf{x}_{\text{rec}}$, and $\mathcal{L}_{\text{CAE}}$
  Update ϕ and γ by Adam algorithm
  $\phi \leftarrow \phi - \mu_1 \partial \mathcal{L}_{\text{CAE}} / \partial \phi$, $\gamma \leftarrow \gamma - \mu_1 \partial \mathcal{L}_{\text{CAE}} / \partial \gamma$
S2: Training of the adversarial cross-modal normalizing flow
Fix network parameters of CAE.
Initialize the parameters $(\theta^{F_{\mathbf{x}}}, \theta^{F_{\mathbf{s}}}, \theta^{D_{\mathbf{x}}}, \theta^{D_{\mathbf{s}}})$ of the networks $F_{\mathbf{x}}, F_{\mathbf{s}}, D_{\mathbf{x}}, D_{\mathbf{s}}$.
Set batch size B_2, learning rate μ_F, μ_D, maximum iteration number T_2, trade-off parameters in Eq. (17)
for t_2 iteration **do**
  Compute $\mathbf{x}_f$ by CAE encoder
  Compute $\widehat{\mathbf{x}_f}, \hat{\mathbf{s}}$, and $\hat{\mathbf{x}}$ by $F_{\mathbf{x}}, F_{\mathbf{s}}$ and CAE decoder
  Compute $\mathcal{L}_{\text{2-Gen}}$
  Update $\theta^{F_{\mathbf{x}}}, \theta^{F_{\mathbf{s}}}$ by Adam algorithm
  $\theta^{F_{\mathbf{x}}} \leftarrow \theta^{F_{\mathbf{x}}} - \mu_F \partial \mathcal{L}_{\text{2-Gen}} / \partial \theta^{F_{\mathbf{x}}}$
  $\theta^{F_{\mathbf{s}}} \leftarrow \theta^{F_{\mathbf{s}}} - \mu_F \partial \mathcal{L}_{\text{2-Gen}} / \partial \theta^{F_{\mathbf{s}}}$
  Fix the network parameters of $F_{\mathbf{x}}, F_{\mathbf{s}}$ and compute $\widehat{\mathbf{x}_f}, \hat{\mathbf{s}}$
  Compute $\mathcal{L}_{\text{2-Dis}}$
  Update $\theta^{D_{\mathbf{x}}}, \theta^{D_{\mathbf{s}}}$ by Adam algorithm
  $\theta^{D_{\mathbf{x}}} \leftarrow \theta^{D_{\mathbf{x}}} - \mu_D \partial \mathcal{L}_{\text{2-Dis}} / \partial \theta^{D_{\mathbf{x}}}$
  $\theta^{D_{\mathbf{s}}} \leftarrow \theta^{D_{\mathbf{s}}} - \mu_D \partial \mathcal{L}_{\text{2-Dis}} / \partial \theta^{D_{\mathbf{s}}}$

discriminators $D_{\mathbf{x}}, D_{\mathbf{s}}$ are updated alternately. The objectives of the generators and the discriminators are

$$\begin{aligned} \min \quad \mathcal{L}_{2-\text{Gen}} =& \lambda_{\text{MLE}} \mathcal{L}_{\text{MLE}} + \lambda_G \mathcal{L}_G \\ &+ \lambda_{\mathbf{x}} \mathcal{L}_{\mathbf{x}} + \lambda_{\mathbf{x}_f} \mathcal{L}_{\mathbf{x}_f} + \lambda_{\mathbf{s}} L_{\mathbf{s}} \\ &+ \lambda_{\mathbf{z}} \mathcal{L}_{\mathbf{z}} + \lambda_{\text{RSA}} \mathcal{L}_{\text{RSA}} + \lambda_{\text{JC}} \mathcal{L}_{\text{JC}}, \end{aligned} \tag{17}$$

$$\min \quad \mathcal{L}_{2-\text{Dis}} = \mathcal{L}_D, \tag{18}$$

$$\text{with} \quad \mathcal{L}_{\text{MLE}} = \mathcal{L}_{\text{MLE}_{\mathbf{x}}} + \mathcal{L}_{\text{MLE}_{\mathbf{s}}}, \quad \mathcal{L}_G = \mathcal{L}_{G_{\mathbf{x}}} + \mathcal{L}_{G_{\mathbf{s}}},$$

where multiple losses are weighted by trade-off parameters.

IV. Experiment

A. Datasets

To validate the generalization of our model, we conduct experiments on two kinds of neural signals. They are the multielectrode recordings of salamander RGCs and the fMRI of the human visual cortex. We use two publicly available datasets[1] in [29] containing the salamander RGC spike signals triggered by natural movie clips and one public dataset[2] in [44] recording fMRI of the human visual cortex triggered by handwritten digits.

The RGC spike trains were collected from isolated salamander retinas with natural movie stimuli. Movie clips were shown to the RGCs at a frame rate of 30 Hz. Fig. 3 shows a demonstration of the data collection. RGCs trigger action potentials in response to the visual stimuli and produce a series of discrete spike trains. They are summed up to spike counts in bins of 1000/30 ms and averaged across different trials for each stimulus. Finally, the continuous neural signals are obtained. The datasets are named according to the stimuli kinds. The detailed information is as follows.

1) Natural Movie-I has 1800 natural gray frames in total. Neural spike trains of 90 RGCs were recorded.
2) Natural Movie-II has 1600 natural gray frames in total. Neural spike trains of 49 RGCs were recorded.

The fMRI dataset from [44] contains the fMRI of one human participant presented with grayscale handwritten digits (numbers 6 and 9). 50 handwritten 6s and 50 handwritten 9s were presented to the subject. Over 3000 voxels from V1, V2, and V3 areas were recorded. Additional information about the data collection can be found in [44]. [7] discarded the unrelated voxels that obtained negative average prediction accuracy prior to the experiments. For the sake of fairness, we use the 1813 voxels selected by [7].

The properties of these three datasets are listed in Table II. The image resolution of all datasets is set as 64×64 for training convenience. For fast convergence, we standardize the pixels of all stimuli to $[0, 1]$ and the neural signals to $[-1, 1]$. The image features are limited within $[-1, 1]$ by the hyperbolic tangent activation function (Tanh).

TABLE II: Properties of three datasets used in the experiments.

Datasets	Recording Methods	ROIs	Cells/ Voxels	Resolution	Instances	Train/test split
Natural Movie-I	Spike Trains	RGCs	90	64×64	1800	10% hold-out (Five times)
Natural Movie-II	Spike Trains	RGCs	49	64×64	1600	10% hold-out (Five times)
Handwritten Digits	fMRI	V1, V2, V3	1813	64×64	100	10-fold cross-validation

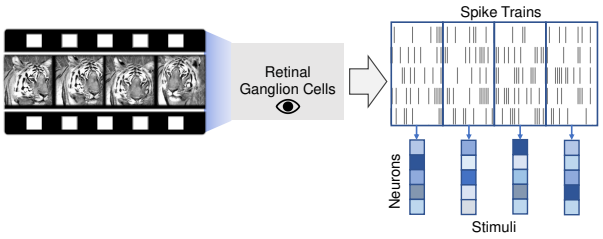

Fig. 3: Demonstration of the neural data collection.

B. Experimental Settings

1) Compared Methods:

- LNP [11]: LNP is a classical neural spike encoding model containing a linear layer and an exponential nonlinear layer.
- CNN-Enc[3] [1]: This is the state-of-the-art RGC spike encoding model. CNN is utilized to mimic the neural circuits in RGCs. We tune the network hyper-parameters in [1] to achieve better performance.
- fwRF[4] [3]: This is the state-of-the-art fMRI encoding model. It weights image features by the receptive fields estimated for every voxel and regresses them to the recorded voxel responses.

[1]https://datadryad.org/stash/dataset/doi:10.5061/dryad.4ch10
[2]http://hdl.handle.net/11633/di.dcc.DSC_2018.00112_485
[3]https://github.com/baccuslab/deep-retina
[4]https://github.com/styvesg/fwrf

TABLE III: Evaluation of neural encoding and decoding performance on test sets of RGC datasets. The mean and standard deviation of each metric are reported. The optimal value on each metric is highlighted. The up/down arrow mark next to each evaluation metric indicates that the larger/smaller the value, the better.

Dataset	Method	Neural Encoding			Neural Decoding		
		$\text{MSE}_{\mathbf{s}}\downarrow$	NLL $\downarrow$	PCC $\uparrow$	$\text{MSE}_{\mathbf{x}}\downarrow$	LPIPS $\downarrow$	SSIM $\uparrow$
Natural Movie-I	LNP	$.072 \pm .003$	$.297 \pm .005$	$.011 \pm .012$			
	CNN-Enc	$.030 \pm .003$	$\mathbf{.174 \pm .004}$	$.770 \pm .015$			
	SID				$\mathbf{.009 \pm .001}$	$.384 \pm .004$	$\mathbf{.744 \pm .007}$
	DCCAE	$.051 \pm .004$	$.718 \pm .042$	$.530 \pm .026$	$.027 \pm .002$	$.573 \pm .016$	$.489 \pm .014$
	CDDG	$.029 \pm .003$	$.448 \pm .024$	$.763 \pm .015$	$.013 \pm .001$	$.375 \pm .003$	$.698 \pm .006$
	CycleGAN	$.058 \pm .006$	$.833 \pm .027$	$.576 \pm .023$	$.013 \pm .000$	$.395 \pm .006$	$.670 \pm .003$
	FLIG	$\mathbf{.028 \pm .003}$	$.197 \pm .004$	$\mathbf{.774 \pm .011}$	$.014 \pm .001$	$\mathbf{.353 \pm .005}$	$.708 \pm .008$
Natural Movie-II	LNP	$.011 \pm .002$	$.140 \pm .009$	$.012 \pm .011$			
	CNN-Enc	$.006 \pm .001$	$\mathbf{.097 \pm .007}$	$.760 \pm .029$			
	SID				$\mathbf{.035 \pm .001}$	$.721 \pm .017$	$.396 \pm .013$
	DCCAE	$.006 \pm .001$	$.270 \pm .032$	$.672 \pm .033$	$.051 \pm .001$	$.587 \pm .022$	$.235 \pm .022$
	CDDG	$.005 \pm .002$	$.243 \pm .049$	$\mathbf{.816 \pm .012}$	$.040 \pm .001$	$.649 \pm .019$	$.324 \pm .016$
	CycleGAN	$.011 \pm .002$	$.462 \pm .066$	$.341 \pm .081$	$.037 \pm .001$	$.686 \pm .015$	$.345 \pm .012$
	FLIG	$\mathbf{.005 \pm .001}$	$.106 \pm .007$	$.767 \pm .030$	$.041 \pm .002$	$\mathbf{.582 \pm .011}$	$\mathbf{.404 \pm .011}$

TABLE IV: Evaluation of neural encoding and decoding performance on the test set of the fMRI dataset. The mean and standard deviation of each metric are reported. The optimal value on each metric is highlighted. The up/down arrow mark next to each evaluation metric indicates that the larger/smaller the value, the better.

Method	Neural Encoding		Neural Decoding		
	$\text{MSE}_{\mathbf{s}}\downarrow$	PCC $\uparrow$	$\text{MSE}_{\mathbf{x}}\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
fwRF	$.193 \pm .075$	$\mathbf{.296 \pm .043}$			
DGMM			$.035 \pm .006$	$15.090 \pm .616$	$.740 \pm .031$
SID			$.036 \pm .008$	$15.001 \pm .905$	$.740 \pm .038$
DCCAE	$.231 \pm .080$	$.199 \pm .042$	$.040 \pm .007$	$14.388 \pm .616$	$.638 \pm .038$
CDDG	$.327 \pm .075$	$.191 \pm .021$	$\mathbf{.030 \pm .005}$	$\mathbf{15.753 \pm .693}$	$.720 \pm .033$
CycleGAN	$.213 \pm .077$	$.186 \pm .044$	$.039 \pm .007$	$14.519 \pm .774$	$.726 \pm .036$
FLIG	$\mathbf{.179 \pm .069}$	$.204 \pm .031$	$.030 \pm .006$	$15.699 \pm .738$	$\mathbf{.759 \pm .032}$

- SID[5] [8]: This is the state-of-the-art RGC spike decoding model. SID projects neural signals into image space through a fully-connected network and then reconstructs stimuli by an auto-encoder.
- DCCAE[6] [35]: It is a deep-learning version of canonical correlation analysis and can be applied in the cross-modal generation.
- CDDG [23]: It achieves simultaneous neural encoding and decoding with JMVAE. CDDG reduces the solution space of the cross-modal generation by imposing constraints on cycle-consistency losses.
- CycleGAN[7] [27]: The model consists of two generators and two discriminators to perform bi-directional generation under cycle-consistency constraints.
- Deep Generative Multiview Model (DGMM)[8] [7]: By using two view-specific generators with a shared latent space, this model builds statistical relationships between visual stimuli and corresponding fMRIs.

2) Training Settings: We cascade layers of Real-NVP together to enhance the representation capability. The network structure and hyper-parameters are tuned carefully on the validation set split from the training set. The detailed network structures and hyper-parameter settings for the three datasets are shown in Appendix B.

On two RGC datasets, we repeatedly ran the program five times, randomly splitting training (90%) and test (10%) sets each time. For the fMRI dataset, we perform a 10-fold cross-validation. Each fold maintains a balance of classes. The mean and standard deviation of each metric are reported.

C. Evaluation Protocol

1) Neural Encoding:

- MSE:

$$\text{MSE}(\hat{\mathbf{s}}, \mathbf{s}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij})^2. \tag{19}$$

- Negative Log-Likelihood (NLL):

$$\text{NLL}(\hat{\mathbf{s}}, \mathbf{s}) = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\hat{\mathbf{s}}_{ij} - \mathbf{s}_{ij} \log \hat{\mathbf{s}}_{ij}). \tag{20}$$

  It reflects the fitting performance of spike counts that are generated through Poisson processes. NLL reaches its minima when $\hat{\mathbf{s}} = \mathbf{s}$.
- Pearson Correlation Coefficient (PCC): MSE reflects the point-to-point error, while PCC reflects the linear correlation. $\mathbf{s}_{:j}$ is the response vector of the j-th neuron to all the stimuli. The PCC of the j-th neuron is

$$\rho(\hat{\mathbf{s}}_{:j}, \mathbf{s}_{:j}) = \frac{\text{cov}(\hat{\mathbf{s}}_{:j}, \mathbf{s}_{:j})}{\sigma_{\hat{\mathbf{s}}_{:j}} \sigma_{\mathbf{s}_{:j}}} \tag{21}$$

  where the numerator is the covariance and the denominator is the product of the standard deviations. The average of all neurons or voxels is used to evaluate the model performance on the held-out sets.

2) Neural Decoding:

- MSE: The MSE of the i-th image pair $\langle \hat{\mathbf{x}}_i, \mathbf{x}_i \rangle$ is

$$\text{MSE}(\hat{\mathbf{x}}_i, \mathbf{x}_i) = \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{k=1}^{W} (\hat{\mathbf{x}}_i^{jk} - \mathbf{x}_i^{jk})^2 \tag{22}$$

  where H, W are image size.

[5]https://github.com/jiankliu/Spike-Image-Decoder
[6]https://github.com/VahidooX/DeepCCA
[7]https://github.com/aitorzip/PyTorch-CycleGAN
[8]https://github.com/ChangdeDu/DGMM

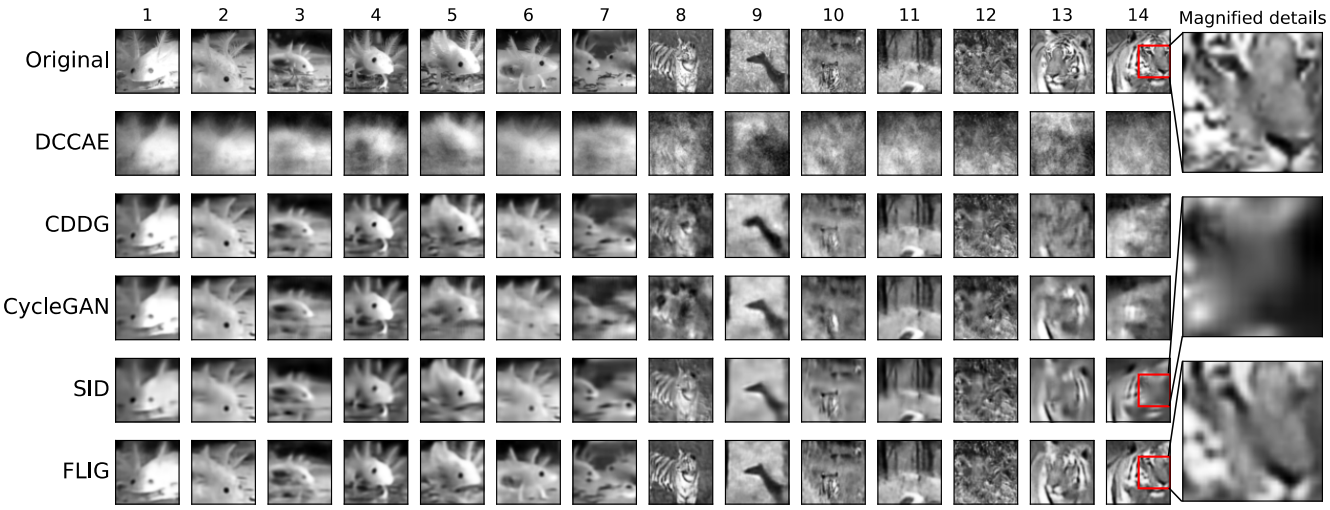

Fig. 4: Examples of decoding results on RGC datasets with FLIG and compared methods. The first row presents the original visual stimuli, of which the first seven columns are from the Natural Movie-I dataset and the last seven columns are from the Natural Movie-II dataset. The following five rows show the corresponding reconstruction results of different methods. On the right, magnified details of the fourteenth example indicate that FLIG preserves more details than SID.

- Peak Signal to Noise Ratio (PSNR): The PSNR of the i-th image pair $\langle \hat{\mathbf{x}}_i, \mathbf{x}_i \rangle$ is

$$\text{PSNR}(\hat{\mathbf{x}}_i, \mathbf{x}_i) = 10 \times \log_{10} \frac{\mathbf{x}_m}{\text{MSE}(\hat{\mathbf{x}}_i, \mathbf{x}_i)} \tag{23}$$

  where $\mathbf{x}_m$ is the maximal pixel value.
- Structural-SImilarity Metric (SSIM): This metric compares the reconstruction quality on a higher level [45]. The SSIM of the i-th image pair $\langle \hat{\mathbf{x}}_i, \mathbf{x}_i \rangle$ is

$$\text{SSIM}(\hat{\mathbf{x}}_i, \mathbf{x}_i) = \frac{(2\mu_{\hat{\mathbf{x}}_i}\mu_{\mathbf{x}_i} + c_1)(2\sigma_{\hat{\mathbf{x}}_i\mathbf{x}_i} + c_2)}{(\mu^2_{\hat{\mathbf{x}}_i} + \mu^2_{\mathbf{x}_i} + c_1)(\sigma^2_{\hat{\mathbf{x}}_i} + \sigma^2_{\mathbf{x}_i} + c_2)} \tag{24}$$

  where $\langle \mu_{\hat{\mathbf{x}}_i}, \mu_{\mathbf{x}_i} \rangle$, $\langle \sigma^2_{\hat{\mathbf{x}}_i}, \sigma^2_{\mathbf{x}_i} \rangle$, and $\sigma_{\hat{\mathbf{x}}_i\mathbf{x}_i}$ are the mean, variance and covariance of $\hat{\mathbf{x}}_i, \mathbf{x}_i$, respectively. c_1, c_2 are constants for computational stability.
- Learned Perceptual Image Patch Similarity (LPIPS) [46]: LPIPS calculates the feature distance between two images with a pretrained DNN and has been indicated to outperform MSE and SSIM in perceptual similarity judgments. Here, we utilize LPIPS with AlexNet [47] as one of the metrics for the neural decoding of natural images.

The average of all image pairs of each indicator is used to evaluate the model performance on the held-out sets.

D. Encoding Performance

The neural encoding performance of different methods on two RGC datasets is summarized in Table III. It can be seen that FLIG matches the state-of-the-art encoding method CNN-Enc. Our method obtains the lowest MSE, while CNN-Enc achieves the lowest NLL on both datasets. This result is reasonable since our method and CNN-Enc optimize neural encoding using the MSE loss and the NLL loss, respectively, and both models are objective-oriented. In addition, FLIG obtains a higher PCC than CNN-Enc and surpasses the other cross-modal generative models (i.e., DCCAE, CDDG, and CycleGAN) on almost all metrics, demonstrating the validity of the flow-based invertible generators. LNP, the traditional RGC encoding model, is not competitive with the other models. The result is consistent with previous studies showing that predicting spike responses triggered by natural stimuli is an intractable task for LNP.

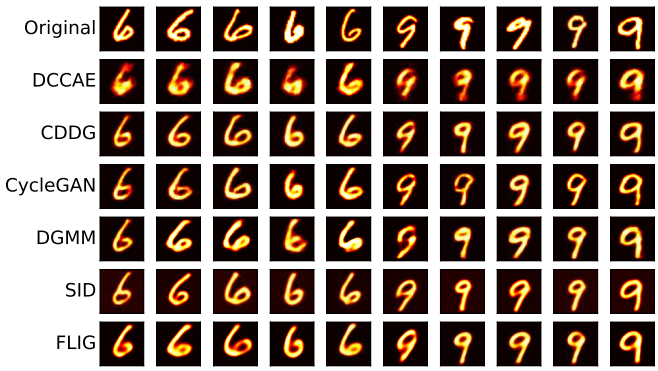

Fig. 5: Decoding results on the handwritten digits datasets with FLIG and compared methods. The first row places the original stimuli. The following six rows show the corresponding reconstruction results of different methods.

Table IV shows the performance of various neural encoding methods on the fMRI dataset. FLIG performs comparably to the state-of-the-art voxel-wise encoding model fwRF. While fwRF has the highest PCC, our method achieves the lowest MSE. Note that our method is more efficient than fwRF because FLIG can synchronously predict the neural responses of all voxels, whereas fwRF performs voxel-wise encoding. In addition, FLIG surpasses all the other cross-modal generation models in neural encoding.

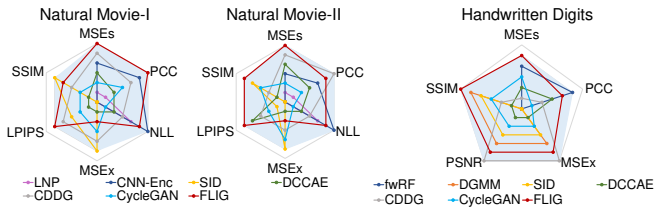

Fig. 6: Radar charts to visualize the comprehensive abilities of all methods on all datasets (best viewed in color). The blue background represents the full score.

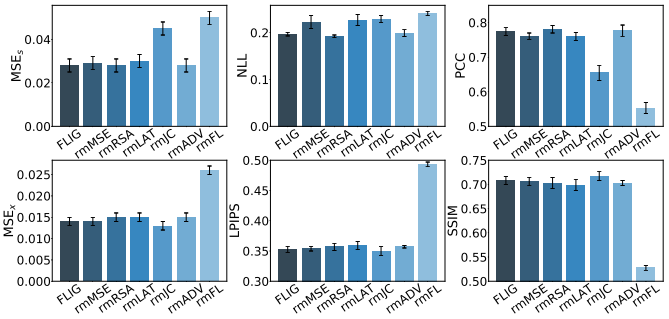

Fig. 7: The results of the ablation experiment on the Natural Movie-I dataset.

E. Decoding Performance

The evaluation of neural decoding is divided into objective assessment and subjective perception. The objective assessment is shown in Table III and Table IV. On the RGC datasets, our model achieves decoding results comparable to those of the state-of-the-art decoding method SID and further outperforms all the other cross-modal models. Interestingly, FLIG outpaces all the other methods in LPIPS, in addition to the competitive results in MSE and SSIM. Based on the prior research and the following qualitative results, it is argued that the MSE and SSIM cannot fully quantify the similarity between two natural images compared to LPIPS. On the human visual cortex dataset, FLIG acquires the highest SSIM among all models.

For the sake of a comprehensive evaluation, we also focus on subjective perception. The decoding results on the RGC and fMRI datasets are demonstrated in Fig. 4 and Fig. 5, respectively. In Fig. 4, our method reconstructs remarkably more details than SID and other methods, such as the tiger facial features in the last two columns of Fig. 4. Some details are magnified and shown on the right of the figure. The results reveal that our flow-based method preserves more information during the generation process. In addition, the subjective perception results also suggest that the LPIPS metric used in Table III is more consistent with human visual perception than the MSE and SSIM metrics. Fig. 5 presents the decoding results on the handwritten digits dataset. FLIG matches SID and CDDG, and surpasses the other methods with more coherent reconstruction. We also visualize the comprehensive performance of the models using radar charts in Fig. 6. The charts are created by transforming the ranks of objective metrics in Table III and Table IV of all methods into scores. The better the metric, the higher the score. If a model ranks first, it obtains the highest score. The blue background denotes a perfect score. The corresponding score will be zero if one model cannot be applied to neural encoding or decoding. FLIG scores 28/33, 28/33, and 25/28 on the Natural Movie-I, Natural Movie-II, and Handwritten Digits datasets, respectively, ranking first. The second-highest scores are 23, 22, and 17, obtained by CDDG. In summary, our proposed method achieves the best comprehensive performance among all compared methods and can be generalized to different neural signals.

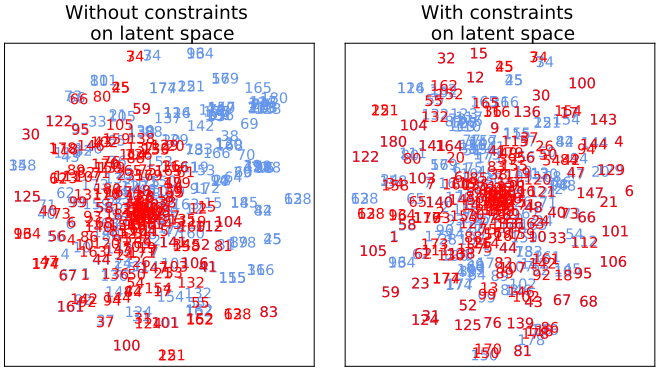

Fig. 8: Visualization of the distribution of $\mathbf{z_x}$ and $\mathbf{z_s}$ using t-SNE (best viewed in color), with samples of the testing set of the Natural Movie-I dataset. $\mathbf{z_x}$ and $\mathbf{z_s}$ are in blue and red, respectively. The numbers are the sample indices.

F. Ablation Experiments

We conduct a series of ablation experiments on the Natural Movie-I dataset to confirm the necessity of the flow-based invertible generators, discriminators, constraints on the latent space, and Jacobian Clamping regularization. Specifically, we observe the effect of removing a particular module or constraint from the model. The related result is shown in Fig. 7. The marks rmMSE, rmRSA, rmLAT, rmJC, and rmADV, represent the methods removing the MSE loss, the RSA-based loss, both the MSE and RSA-based losses on the latent space, Jacobian Clamping, and adversarial learning (i.e., removing the discriminators), respectively. The mark rmFL refers to the method that replaces the normalizing flows $F_{\mathbf{x}}$ and $F_{\mathbf{s}}$ with fully-connected networks. The abolishment of the MSE constraint deteriorates both the encoding and decoding results. The removal of the RSA-based loss degrades the decoding performance. The situation gets worse when both restrictions are disabled. Furthermore, Fig. 8 provides a t-SNE visualization [48] of the distributions of $\mathbf{z_x}$ and $\mathbf{z_s}$ with and without the constraints on latent space. Each blue number and each red number represent $\mathbf{z}_{\mathbf{x}_i}$ and $\mathbf{z}_{\mathbf{s}_i}$ in the testing set of the Natural Movie-I dataset, respectively. The numbers are the sample indices. The results suggest that constraints on latent space could lead to better cross-modal feature alignment. The quantitative and qualitative results imply that latent space constraints are indispensable for the generation of two heterogeneous modalities.

The decoding performance of the model without Jacobian Clamping regularization is superior to that of the original model, while the encoding performance is the worst of all cases except for rmFL. These changes highlight the importance of the Jacobian Clamping regularization for the synchronous convergence of neural encoding and decoding. Without discriminators, neural decoding performance deteriorates, reflecting that the generation of high-dimensional data such as images is more dependent on adversarial learning. Finally, the encoding and decoding performance of the model without flow-based generators dramatically gets worse, manifesting the effectiveness of the normalizing flows.

V. Discussion

The proposed flow-based generative model integrates visual neural encoding and decoding into one framework. Furthermore, it is not limited to particular types of neural signals. Here, we discuss the two-stage design and the strengths and limitations of our model.

A. Two-stage design of FLIG

Due to the large modal gap between neural signals and visual stimuli, it is not easy to learn a direct mapping from neural data to images [20]. FLIG alleviates the issue by projecting the two modalities onto an intermediate feature space and imposing constraints to align their representations on such a space. In addition, FLIG adopts two-stage training rather than end-to-end training because the latter requires learning more parameters at once and is more data-dependent [49]. We adopt a two-stage training procedure to prevent model overfitting to the training sets.

B. Advantages of FLIG

The first advantage of FLIG is the lossless generation, which is enabled by normalizing flows. In comparison to irreversible networks such as VAE and GAN, the bijective transformations of normalizing flows minimize information losses during generation. Despite the same-dimensional latent spaces of the three models, the comprehensive performance of FLIG is far superior to that of CDDG and CycleGAN, demonstrating the advantage of normalizing flows. In addition, FLIG is naturally good at preserving cycle consistency during cross-modal generation, even without related constraints the CDDG method used.

Second, FLIG can learn neural encoding and decoding simultaneously. This property can not only unify the research on neural encoding and decoding but also save training expenses.

Finally, our proposed method can be taken as a universal cross-modal generation method for two reasons. The first reason is that FLIG breaks through the limitation of normalizing flows and enables the cross-modal generation of two modalities in distinct dimensions. The second reason is that the local and global constraints on the latent spaces can ensure domain alignment between two heterogeneous modalities. Experimental results show the strong generalization ability of our model across different neural signals. Therefore, FLIG can be theoretically applied to the neural encoding and decoding of a variety of neural signals, as well as cross-modal generation of image and text, audio and text. However, the latter is not the focus of this paper.

C. Limitations of FLIG

The invertible mapping of normalizing flows is a double-edged sword. It does not support dimension changes, which may limit its expression ability. A cascade of flows can improve the representation capability. In the future, we will replace Real-NVP with more efficient normalizing flows.

VI. Conclusion

In this paper, we propose a novel neural encoding and decoding method with flow-based invertible generative networks to mitigate information losses during cross-modal generation and consider the dual relationship between neural encoding and decoding. The model training contains two stages. First, an auto-encoder extracts image features. Second, the generation of the visual stimuli and neural signals is conducted by an adversarial cross-modal normalizing flow and the well-trained auto-encoder. Local and global constraints on the latent space shrink the modal gap. Experimental results on two RGC spike datasets and one fMRI dataset of the human visual cortex indicate that the proposed method reconstructs stimuli images with more details than other comparison methods. In addition, our model matches the state-of-the-art models and achieves the best comprehensive performance.

In the future, we plan to apply our model to neural signals recorded by alternative techniques. We also expect to probe the similarities and differences between our flow-based model and visual processing in the human visual cortex.

Appendix A
Proof of the Coupling Inequality

Let $G_{\mathbf{x}\to\mathbf{s}} = F_{\mathbf{s}}F_{\mathbf{x}}^{-1}$. In the following proof, we note $G_{\mathbf{x}\to\mathbf{s}}$ as G for convenience. The condition number $\kappa(G) = \|G\|\|G^{-1}\|$. Then,

$$\begin{aligned}\widehat{\mathbf{x}_f} &= F_{\mathbf{x}}(\mathbf{z}_{\mathbf{s}}) = F_{\mathbf{x}}(F_{\mathbf{s}}^{-1}(\mathbf{s})),\\ \Rightarrow \mathbf{s} &= F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\widehat{\mathbf{x}_f})) = G(\widehat{\mathbf{x}_f}),\\ \hat{\mathbf{s}} &= F_{\mathbf{s}}(\mathbf{z}_{\mathbf{x}}) = F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\mathbf{x}_f)) = G(\mathbf{x}_f).\end{aligned}$$

By plugging the corresponding terms into $\mathcal{L}_{\mathbf{s}}$, we obtain

$$\mathcal{L}_{\mathbf{s}} = \|\hat{\mathbf{s}} - \mathbf{s}\|_2^2 = \|G(\mathbf{x}_f) - G(\widehat{\mathbf{x}_f})\|_2^2, \tag{25}$$

which indicates that the relative trends of $L_{\mathbf{s}}$ and $L_{\mathbf{x}_f}$ are coupled to some extent. Since $G(\mathbf{x}_f) = \hat{\mathbf{s}}, G(\widehat{\mathbf{x}_f}) = \mathbf{s}$, we get

$$\|G(\mathbf{x}_f)\| = \|\hat{\mathbf{s}}\|, \tag{26}$$

$$\|G(\widehat{\mathbf{x}_f})\| = \|\mathbf{s}\|. \tag{27}$$

The matrix norm of a matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\| = \max_{\mathbf{y}} \frac{\|\mathbf{A}\mathbf{y}\|}{\|\mathbf{y}\|} \tag{28}$$

for any $\mathbf{y} \neq \mathbf{0}$. It can be deduced that

$$\|\mathbf{A}^{-1}\| = \max_{\mathbf{v}} \frac{\|\mathbf{A}^{-1}\mathbf{v}\|}{\|\mathbf{v}\|} = 1/\min_{\mathbf{v}} \frac{\|\mathbf{v}\|}{\|\mathbf{A}^{-1}\mathbf{v}\|} = 1/\min_{\mathbf{y}} \frac{\|\mathbf{A}\mathbf{y}\|}{\|\mathbf{y}\|} \tag{29}$$

with $\mathbf{y} = \mathbf{A}^{-1}\ \mathbf{v}$. Substituting the matrix and the vector in Eq. (28) and Eq. (29) with G and $\mathbf{x}_f$ or $\widehat{\mathbf{x}_f}$, we can obtain

$$\|G\| \geq \frac{\|G(\mathbf{x}_f)\|}{\|\mathbf{x}_f\|}, \tag{30}$$

$$\|G^{-1}\| \geq 1/\frac{\|G(\mathbf{x}_f)\|}{\|\mathbf{x}_f\|}, \tag{31}$$

$$\|G\| \geq \frac{\|G(\widehat{\mathbf{x}_f})\|}{\|\widehat{\mathbf{x}_f}\|}, \tag{32}$$

$$\|G^{-1}\| \geq 1/\frac{\|G(\widehat{\mathbf{x}_f})\|}{\|\widehat{\mathbf{x}_f}\|}. \tag{33}$$

Combining Eq. (27) and Eq. (32) together, we get

$$\|G\|\|\widehat{\mathbf{x}_f}\| \geq \|G(\widehat{\mathbf{x}_f})\| = \|\mathbf{s}\|. \tag{34}$$

Combining Eq. (26) and Eq. (31) together, we obtain

$$\|G^{-1}\|\|G(\mathbf{x}_f)\| = \|G^{-1}\|\|\hat{\mathbf{s}}\| \geq \|\mathbf{x}_f\|. \tag{35}$$

Multiplying the above two inequalities, we have

$$\begin{aligned} \|G\|\|G^{-1}\|\|\widehat{\mathbf{x}_f}\|\|\hat{\mathbf{s}}\| &\geq \|\mathbf{x}_f\|\|\mathbf{s}\| \\ \Rightarrow \frac{\|\mathbf{s}\|}{\|\hat{\mathbf{s}}\|} &\leq \kappa(G)\frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|}. \end{aligned} \tag{36}$$

In a similar way, we can prove the other side of Eq. (15). Combining Eq. (30) and Eq. (26) together, we get

$$\|G\|\|\mathbf{x}_f\| \geq \|G(\mathbf{x}_f)\| = \|\hat{\mathbf{s}}\|. \tag{37}$$

Combining Eq. (33) and Eq. (27) together, we obtain

$$\|G^{-1}\|\|G(\widehat{\mathbf{x}_f})\| = \|G^{-1}\|\|\mathbf{s}\| \geq \|\widehat{\mathbf{x}_f}\|. \tag{38}$$

Multiplying the above two inequalities, we have

$$\begin{aligned} \|G\|\|G^{-1}\|\|\mathbf{x}_f\|\|\mathbf{s}\| &\geq \|\widehat{\mathbf{x}_f}\|\|\hat{\mathbf{s}}\| \\ \Rightarrow \frac{\|\mathbf{s}\|}{\|\hat{\mathbf{s}}\|} &\geq \frac{1}{\kappa(G)}\frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|}. \end{aligned} \tag{39}$$

Eventually, we prove that

$$\frac{1}{\kappa(G_{\mathbf{x}\to\mathbf{s}})}\frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|} \leq \frac{\|\mathbf{s}\|}{\|\hat{\mathbf{s}}\|} \leq \kappa(G_{\mathbf{x}\to\mathbf{s}})\frac{\|\widehat{\mathbf{x}_f}\|}{\|\mathbf{x}_f\|}. \tag{40}$$

APPENDIX B
NETWORK ARCHITECTURES AND HYPER-PARAMETER SETTINGS

Table V shows information about the CAE architecture used in the experiments. In the table, the abbreviations indicate the convolution (Conv), batch normalization (Batchnorm), upsampling (Upsample) operations, and the fully-connected layer (FC). The numbers before Conv denote the kernel size, stride size, padding size, and the output channel number of the convolution operation. For instance, **(7, 2, 3, 64) Conv** means the convolution with a kernel size of 7×7, 2 strides, 3 paddings and a 64-channel output. Similarly, the number before FC denotes the output dimension of the fully-connected layer. The architecture can be generalized to three datasets by setting the latent dimension to M, the dimension of the corresponding neural signals.

TABLE V: CAE architecture used in the experiments. The latent dimension equals the dimension M of the corresponding neural signals.

Encoder $Enc_{\mathbf{x}}$	**Decoder** $Dec_{\mathbf{x}}$
Input 64×64 image	Input $\in R^M$
(7, 2, 3, 64) Conv. Batchnorm. ReLU	$(4 \times 4 \times 256)$ FC.
(5, 2, 2, 128) Conv. Batchnorm. ReLU. Dropout.	$\times 2$ Upsample. (3, 1, 1, 256) Conv. Batchnorm. ReLU. Dropout.
(3, 2, 1, 256) Conv. Batchnorm. ReLU. Dropout.	$\times 2$ Upsample. (3, 1, 1, 128) Conv. Batchnorm. ReLU. Dropout.
(3, 2, 1, 256) Conv. Batchnorm. ReLU. Dropout.	$\times 2$ Upsample. (5, 1, 2, 64) Conv. Batchnorm. ReLU. Dropout.
(M) FC. Tanh.	$\times 2$ Upsample. (7, 1, 3, 1) Conv. Sigmoid.

The architecture of the normalizing flow modules, i.e., $F_{\mathbf{x}}$ and $F_{\mathbf{s}}$, is the stack of several basic flow units. Each basic flow unit is based on the scale and translation functions, $s(\cdot)$ and $t(\cdot)$, as shown in Eq. (5). A basic flow unit is composed of a five-layer fully-connected network ($d \to 128 \to 128 \to 128 \to M - d$), where d is the dimension of the first slice of the intact input (See Section III-C). The activation function of the hidden layers is Tanh. The numbers of basic flow units of $F_{\mathbf{x}}$ and $F_{\mathbf{s}}$ are 15 and 1, respectively.

The discriminators, $D_{\mathbf{x}}$ and $D_{\mathbf{s}}$, used in the experiments are three-layer fully-connected networks ($M \to \frac{M}{2} \to 1$) and output a value between 0 and 1 using Sigmoid activation function. To ensure more stable training, we give the discriminators the gradient penalty proposed in [41]. Concretely,

$$\begin{aligned} \mathcal{L}_{D_{gp-\mathbf{x}}} &= \mathbb{E}_{\mathbf{s}\sim p_{\text{data}}(\mathbf{s})}[(\|\nabla_{\widehat{\mathbf{x}_f}} D_{\mathbf{x}}(\widehat{\mathbf{x}_f})\|_2 - 1)^2], \\ \mathcal{L}_{D_{gp-\mathbf{s}}} &= \mathbb{E}_{\mathbf{x}_f\sim p_{\text{data}}(\mathbf{x}_f)}[(\|\nabla_{\hat{\mathbf{s}}} D_{\mathbf{s}}(\hat{\mathbf{s}})\|_2 - 1)^2], \\ \text{with} \quad \widehat{\mathbf{x}_f} &= F_{\mathbf{x}}(F_{\mathbf{s}}^{-1}(\mathbf{s})), \quad \hat{\mathbf{s}} = F_{\mathbf{s}}(F_{\mathbf{x}}^{-1}(\mathbf{x}_f)). \end{aligned}$$

The trade-off parameters λ_{MLE}, λ_G, $\lambda_{\mathbf{x}}$, $\lambda_{\mathbf{x}_f}$, $\lambda_{\mathbf{z}}$, λ_{RSA}, and λ_{JC} in Eq. (17) are set to 0.01, 0.01, 100, 100, 10, 1, and 10, except that $\lambda_{\mathbf{s}}$ is set to 100, 10 and 200 on the Natural-Movie I, Natural Movie-II and Handwritten Digits datasets. In addition, (λ^+, λ^-) is set to $(0.1, 0.05)$, $(0.4, 0.2)$, and $(0.5, 0)$ on the Natural-Movie I, Natural Movie-II, and Handwritten Digits datasets. CAE's learning rate is 1e-5. The normalizing flows and the discriminators learn at a rate of 5e-5 on the Natural-Movie I and Natural Movie-II datasets and at a rate of 1e-5 on the Handwritten Digits dataset. The Adam optimizer is used for all updates.

REFERENCES

[1] L. McIntosh, N. Maheswaranathan, A. Nayebi, S. Ganguli, and S. Baccus, "Deep learning models of the retinal response to natural scenes," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 29, 2016, pp. 1369–1377.

[2] E. Batty, J. Merel, N. Brackbill, A. Heitman, A. Sher, A. Litke, E. Chichilnisky, and L. Paninski, "Multilayer recurrent network models of primate retinal ganglion cell responses," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2017.

[3] G. Styves and T. Naselaris, "The feature-weighted receptive field: an interpretable encoding model for complex feature spaces," *NeuroImage*, vol. 180, pp. 188–202, 2018.

[4] K. Kay, T. Naselaris, R. J. Prenger, and J. L. Gallant, "Identifying natural images from human brain activity," *Nature*, vol. 452, no. 7185, pp. 352–355, 2008.

[5] T. Naselaris, R. J. Prenger, K. Kay, M. Oliver, and J. L. Gallant, "Bayesian reconstruction of natural images from human brain activity," *Neuron*, vol. 63, no. 6, pp. 902–915, 2009.

[6] D. Li, C. Du, and H. He, "Semi-supervised cross-modal image generation with generative adversarial networks," *Pattern Recognition*, vol. 100, p. 107085, 2020.

[7] C. Du, C. Du, L. Huang, and H. He, "Reconstructing perceived images from human brain activities with bayesian deep multiview learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 8, pp. 2310–2323, 2018.

[8] Y. Zhang, S. Jia, Y. Zheng, Z. Yu, Y. Tian, S. Ma, T. Huang, and J. K. Liu, "Reconstruction of natural visual scenes from neural spikes with deep neural networks," *Neural Networks*, vol. 125, pp. 19–30, 2020.

[9] T. Siriborvornratanakul, "Through the realities of augmented reality," in *International Conference on Human-Computer Interaction, HCI*, vol. 11786, 2019, pp. 253–264.

[10] R. Beliy, G. Gaziv, A. Hoogi, F. Strappini, T. Golan, and M. Irani, "From voxels to pixels and back: Self-supervision in natural-image reconstruction from fmri," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 32, 2019, pp. 6517–6527.

[11] E. P. Simoncelli, J. W. Pillow, L. Paninski, and O. Schwartz, "Characterization of neural responses with stochastic stimuli," in *The cognitive neurosciences, III*. MIT Press, 2004, pp. 327–338.

[12] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. Chichilnisky, and E. P. Simoncelli, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, no. 7207, pp. 995–999, 2008.

[13] J. K. Liu, H. M. Schreyer, A. Onken, F. Rozenblit, M. H. Khani, V. Krishnamoorthy, S. Panzeri, and T. Gollisch, "Inference of neuronal functional circuitry with spike-triggered non-negative matrix factorization," *Nature Communications*, vol. 8, no. 1, pp. 1–14, 2017.

[14] G. P. Das, P. J. Vance, D. Kerr, S. A. Coleman, T. M. McGinnity, and J. K. Liu, "Computational modelling of salamander retinal ganglion cells using machine learning approaches," *Neurocomputing*, vol. 325, pp. 101–112, 2019.

[15] H. Wen, J. Shi, Y. Zhang, K. 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, 2018.

[16] H. Wang, L. Huang, C. Du, D. Li, B. Wang, and H. He, "Neural encoding for human visual cortex with deep neural networks learning "what" and "where"," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2014.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 27, 2014, pp. 2672–2680.

[19] K. Han, H. Wen, J. Shi, K.-H. Lu, Y. Zhang, D. Fu, and Z. Liu, "Variational autoencoder: An unsupervised model for encoding and decoding fmri activity in visual cortex," *NeuroImage*, vol. 198, pp. 125–136, 2019.

[20] Z. Ren, J. Li, X. Xue, X. Li, F. Yang, Z. Jiao, and X. Gao, "Reconstructing seen image from brain activity by visually-guided cognitive representation and adversarial learning," *NeuroImage*, vol. 228, p. 117602, 2021.

[21] S. Huang, L. Sun, M. Yousefnezhad, M. Wang, and D. Zhang, "Temporal information guided generative adversarial networks for stimuli image reconstruction from human brain activities," *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[22] R. Caruana, "Multitask learning," in *Learning to Learn*, S. Thrun and L. Y. Pratt, Eds. Springer, 1998, pp. 95–133.

[23] Q. Zhou, C. Du, D. Li, H. Wang, J. K. Liu, and H. He, "Simultaneous neural spike encoding and decoding based on cross-modal dual deep generative model," in *International Joint Conference on Neural Networks, IJCNN*. IEEE, 2020, pp. 1–8.

[24] M. Suzuki, K. Nakayama, and Y. Matsuo, "Joint multimodal learning with deep generative models," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2017.

[25] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proceedings of the International Conference on Machine Learning, ICML*, 2015, pp. 1530–1538.

[26] N. Kriegeskorte, M. Mur, and P. A. Bandettini, "Representational similarity analysis-connecting the branches of systems neuroscience," *Frontiers in systems neuroscience*, vol. 2, p. 4, 2008.

[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 2017, pp. 2242–2251.

[28] A. Grover, C. Chute, R. Shu, Z. Cao, and S. Ermon, "Alignflow: Cycle consistent learning from multiple domains via normalizing flows," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 4028–4035.

[29] A. Onken, J. K. Liu, P. C. R. Karunasekara, I. Delis, T. Gollisch, and S. Panzeri, "Using matrix and tensor factorizations for the single-trial analysis of population spike trains," *PLoS computational biology*, vol. 12, no. 11, p. e1005189, 2016.

[30] N. Maheswaranathan, L. T. McIntosh, H. Tanaka, S. Grant, D. B. Kastner, J. B. Melander, A. Nayebi, L. Brezovec, J. Wang, S. Ganguli *et al.*, "The dynamic neural code of the retina for natural scenes," *BioRxiv*, p. 340943, 2019.

[31] Q. Yan, Y. Zheng, S. Jia, Y. Zhang, Z. Yu, F. Chen, Y. Tian, T. Huang, and J. K. Liu, "Revealing fine structures of the retinal receptive field by deep-learning networks," *IEEE transactions on cybernetics*, 2020.

[32] V. Botella-Soler, S. Deny, G. Martius, O. Marre, and G. Tkačik, "Nonlinear decoding of a complex movie from the mammalian retina," *PLoS computational biology*, vol. 14, no. 5, p. e1006057, 2018.

[33] Y. Miyawaki, H. Uchida, O. Yamashita, M. Sato, Y. Morito, H. C. Tanabe, N. Sadato, and Y. Kamitani, "Visual image reconstruction from human brain activity using a combination of multiscale local image decoders," *Neuron*, vol. 60, no. 5, pp. 915–929, 2008.

[34] W. Huang, H. Yan, C. Wang, J. Li, Z. Zuo, J. Zhang, Z. Shen, and H. Chen, "Perception-to-image: Reconstructing natural images from the brain activity of visual perception," *Annals of Biomedical Engineering*, vol. 48, no. 9, pp. 2323–2332, 2020.

[35] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *Proceedings of the International Conference on Machine Learning, ICML*, vol. 37, 2015, pp. 1083–1092.

[36] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, 1936.

[37] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2017.

[38] L. Dinh, D. Krueger, and Y. Bengio, "NICE: non-linear independent components estimation," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2015.

[39] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 32, 2019, pp. 7509–7520.

[40] A. Grover, M. Dhar, and S. Ermon, "Flow-gan: Combining maximum likelihood and adversarial learning in generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[41] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems, NeurIPS*, vol. 30, 2017.

[42] A. Odena, J. Buckman, C. Olsson, T. B. Brown, C. Olah, C. Raffel, and I. Goodfellow, "Is generator conditioning causally related to gan performance?" in *Proceedings of the International Conference on Machine Learning, ICML*, vol. 80, 2018, pp. 3846–3855.

[43] K. B. Datta, *Matrix and linear algebra*. Prentice-Hall of India New Delhi, India, 1991.

[44] M. A. J. V. Gerven, F. P. D. Lange, and T. Heskes, "Neural decoding with hierarchical generative models," *Neural Computation*, vol. 22, no. 12, pp. 3127–3142, 2010.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[46] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 586–595.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 25, 2012, pp. 1097–1105.

[48] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[49] G. Shen, K. Dwivedi, K. Majima, T. Horikawa, and Y. Kamitani, "End-to-end deep image reconstruction from human brain activity," *Frontiers in Computational Neuroscience*, p. 21, 2019.