Faster Convergence in Deep-Predictive-Coding Networks to Learn Deeper Representations

Isaac J. Sledge, Member, IEEE, and José C. Príncipe, Life Fellow, IEEE

Abstract—Deep-predictive-coding networks (DPCNs) are hierarchical, generative models. They rely on feed-forward and feedback connections to modulate latent feature representations of stimuli in a dynamic and context-sensitive manner. A crucial element of DPCNs is a forward-backward inference procedure to uncover sparse, invariant features. However, this inference is a major computational bottleneck. It severely limits the network depth due to learning stagnation. Here, we prove why this bottleneck occurs. We then propose a new forward-inference strategy based on accelerated proximal gradients. This strategy has faster theoretical convergence guarantees than the one used for DPCNs. It overcomes learning stagnation. We also demonstrate that it permits constructing deep and wide predictive-coding networks. Such convolutional networks implement receptive fields that capture well the entire classes of objects on which the networks are trained. This improves the feature representations compared with our lab’s previous nonconvolutional and convolutional DPCNs. It yields unsupervised object recognition that surpass convolutional autoencoders and is on par with convolutional networks trained in a supervised manner.

Index Terms—Autoencoder, bio-inspired vision, predictive coding, unsupervised learning.

I. INTRODUCTION

Predictive coding is a promising theory for unsupervised sensory information processing. Under this theory, a hierarchical, generative model [1], [2] of a dynamic environment is formed. This model is consistently updated to infer possible states of the environment and physical causes of the environmental stimuli. These causes, in turn, permit reproducing the stimuli [3] (see Section II). Several predictive coders have been created and their biological plausibility investigated [1], [4]–[6]. None of these early contributions has been known to form causes that are highly discriminative for complex stimuli, however. Our lab thus developed multistage, deep-predictive-coding networks (DPCNs) [7]–[9]. We showed that DPCNs could form discriminative causes of temporal, spatial, and spatio-temporal stimuli in certain cases. Alternate networks later followed [10], [11].

DPCNs learn about an environment in an unsupervised manner. They can thus be thought of as parameter-light, nontraditional autoencoders. DPCNs differ substantially from autoencoders, though. The former contain feed-forward and recurrent, feedback connections, allowing information to be propagated between stages to stabilize the internal representation. Another distinction is that DPCNs do not have a corresponding decoder. They hence, require a self-organizing principle to be effective. That is, they must learn to extract meaningful features, in the form of causes, that cluster the stimuli effectively. This clustering objective is aided by imposing cause sparsity and transformation invariance. Including these cause constraints make DPCNs generalize well to novel stimuli. Lastly, DPCNs are composed of stages, not layers. Each stage implements a recurrent state model. This permits DPCNs to characterize the dynamics of temporal and spatio-temporal stimuli.

DPCNs exhibit promise for unsupervised object recognition in images and video. Often, convolutional DPCNs like [9] learn sparse, invariant features that are better for classification than those from convolutional autoencoders. Such behavior arises from the interaction of feed-forward and feedback connections in the DPCNs. It also arises due to the implicit supervision imposed from leveraging temporal information [12]. Multiple presentations of the stimuli additionally facilitate the extraction of spatial and temporal regularities [13], [14], which we hypothesize permits high-level object analysis [15], like object recognition. Sparsity contributes too [16], [17], as it aids in generalization and can preempt overfitting to specific stimuli.

Learning sufficiently robust features in a DPCN is quite computationally intensive. A multistage optimization strategy, based on proximal gradients [18], [19], is typically used Chalasani and Príncipe [7]–[9] to conduct feed-forward and feedback inference of the causes. Both forms of inference are needed for cause self-organization. Subquadratic function-value convergence rates are theoretically guaranteed for this strategy [20], [21]. Only sublinear rates are often obtainable, however, due to severe oscillations in the cost. That is, the search is not a pure descent strategy in some cases. This has the dual effect of returning poor causes and doing so slowly.

Due to these optimization difficulties, DPCNs are practically limited to two stages. Two stages may be sufficient for...
characterizing certain stimuli. However, it typically will not yield representations that handle objects in complex environments. The networks exhibit poor stimuli-reconstruction performance when extended beyond two stages due to being stymied by poor convergence [22]. The deeper stages do not reach a stable cause representation, which impacts the causes in preceding stages. Learning essentially stagnates regardless of how many stimuli are presented. This, in turn, prevents object recognition for many challenging environments. The causes simply do not organize the stimuli well. Moreover, the causes are not semantically rich enough to handle distractors. Even simple textural backgrounds in visual stimuli can be sufficiently distracting enough to confound the DPCNs. The causes are also sometimes unable to handle object variability effectively. They cannot always resolve that objects from distinct viewpoints are the same, for instance.

Here, we develop a novel inference process that enables investigators to go beyond the two-stage network limitation that is observed in practice for DPCNs. This leads to what we refer to as accelerated DPCNs (ADPCNs). Both the DPCNs and ADPCNs are unsupervised networks. They share the same underlying architecture, with one exception: the ADPCNs extract convolutional features (see Section III). Both properties mean that ADPCNs are better suited for discrimination than our lab’s original, nonconvolutional DPCNs [7], [8]. The improved inference also makes ADPCNs better than our lab’s convolutional-recurrent-predictive networks (CRPNs) [9]. CRPNs similarly rely on slow proximal-gradient inference.

More specifically, we replace the proximal gradient search in the original DPCN with an accelerated version (see Section III). The accelerated approach in the ADPCN relies on a polynomial inertial sequence for updating the internal feature representations. The inertial sequence has the effect of sufficiently delaying the occurrence of cost oscillation. We prove this (see Appendix A in the supplementary material). Monotonically decreasing costs occur throughout almost all of the learning process. We perform in-place restarts of the inference when it does not. The ADPCNs therefore extract meaningful error signals that stabilize the sparse causes early during training. ADPCNs also possess a faster, subpolynomial rate of function-value convergence compared to the inference scheme used by DPCNs (see Appendix A in the supplementary material). ADPCNs hence, exhibit improved empirical convergence too over DPCNs. We are thus able to efficiently train both deep and wide predictive-coding networks whose learning does not stall. The deep convolutional layers of the ADPCN extract progressively richer, transformation-invariant features for sparsely describing complex stimuli. They aid in stimuli generalization. Wide convolutional layers better approximate interactions between stimuli and causes than narrower ones. They promote some measure of stimuli memorization without overfitting, which permits recognizing perceptually similar objects from the same class. While such behaviors can manifest in convolutional DPCNs, they do not due to the aforementioned slow inference.

We show that the feature representations from the ADPCN are far more robust than those obtainable for DPCNs. Even a single-stage difference between convolutional DPCNs and ADPCNs leads to huge improvements in discriminability. In particular, the later-stage causes in the ADPCNs have receptive fields that embody the entirety of the objects being presented, despite the lack of training labels (see Section IV). We observe this phenomenon across a variety of benchmark datasets. It enables multiclass object recognition under different scene conditions. The ADPCNs can handle perspective changes, shape changes, illumination changes, and more. ADPCNs thus form an approximate identity mapping that preserves perceptual difference. Whole-object sensitivity also yields unsupervised classifiers that are on par with supervised-trained convolutional and convolutional-recurrent networks (see Appendix B in the supplementary material). The ADPCNs have orders of magnitude fewer parameters, though, than these other deep networks. ADPCNs, just like DPCNs, are parameter-light, nontraditional autoencoders.

II. PREDICTIVE CODING

The objective of predictive coding is to approximate external sensory stimuli using generative, latent-variable models. Such models hierarchically encode residual prediction errors. The prediction errors are the differences between either the actual stimuli or a transformed version of it and the predicted stimuli produced from the underlying latent variables. We refer to these latent variables as causes. We also interchangeably refer to causes as features.

By learning in this way, the internal representations of a predictive-coding model are modified only for unexpected changes in the stimuli. This enables a model to recall stimuli that it has encountered before. It also enables the model to adapt to new stimuli without disrupting its internal representation of the environment for previously observed stimuli.

We can characterize predictive coding in the following way for temporal and spatio-temporal stimuli. Examples include audio and video, respectively. Both DPCNs and ADPCNs are instances of this general framework.

Definition 1 (Predictive Coding Model): Let \( y_t \in \mathbb{R}^p \) represent a time-varying sensory stimulus at time \( t \). The stimuli can be described by an underlying cause, \( \kappa_t \in \mathbb{R}^d \), and a time-varying intermediate state, \( \gamma_{1,t} \in \mathbb{R}^{d_1} \), through a pair of \( \theta \)-parameterized mapping functions, \( f_1 : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^p \), the cause-update function, and \( g_1 : \mathbb{R}^k \times \mathbb{R}^{d_1} \rightarrow \mathbb{R}^k \), the state-transition function. These functions define a latent-variable model

\[
y_t = f_1(\gamma_{1,t}; \theta) + \epsilon_{1,t}, \quad \gamma_{1,t} = g_1(\gamma_{1,t-1}, \kappa_t; \theta) + \epsilon_{0,t}.
\]

Here, \( \epsilon_{1,t} \in \mathbb{R}^p \) and \( \epsilon_{0,t} \in \mathbb{R}^k \) are noise terms that represent the stochastic and model uncertainty, respectively, in the predictions. This model can be extended to a multistage hierarchy by cascading additional \( \theta \)-parameterized mapping functions, \( f_i : \mathbb{R}^{d_{i-1}} \rightarrow \mathbb{R}^{d_i} \) and \( g_i : \mathbb{R}^{k_i} \times \mathbb{R}^{d_{i-1}} \rightarrow \mathbb{R}^{d_i} \), at each stage \( i \) beyond the first

\[
\kappa_{i-1,t} = f_i(\gamma_{i-1,t}; \theta) + \epsilon_{i,t}, \quad \gamma_{i,t} = g_i(\gamma_{i-1,t}, \kappa_t; \theta) + \epsilon_{i-1,t}.
\]

Here, \( \kappa_{i,t} \in \mathbb{R}^{d_i}, \gamma_{i,t} \in \mathbb{R}^{d_i}, \epsilon_{i,t} \in \mathbb{R}_{d_{i-1}} \) and \( \epsilon_{i-1,t} \in \mathbb{R}^{d_i} \). Spatial stimuli, which include images, are a degenerate case of this framework. They have no temporal component, so there is no modification of the states as they are fed back. States are still extracted, though. Predictive coding thus behaves similar to sparse coding [23] in this case. The main difference is that, for predictive coding, the states are pooled and transformed to form causes. Sparse coding only has notions of states. For DPCNs and ADPCNs, the causes are made invariant, which aids in discrimination. Feature invariance does not naturally occur in many sparse coding and hierarchical sparse coding models [24], [25].
For the above framework, both feed-forward, bottom-up, and feedback, top-down processes are used to characterize observed stimuli. For the feed-forward case, the observed stimuli are propagated through the model to extract progressively abstract details. The stimuli are first converted to a series of states that encode either spatial, temporal, or spatio-temporal relationships. The type of relationship depends on the stimuli being considered. These states are then made invariant to various transformations, thereby forming hidden causes. The causes are latent variables that describe the environment, as we noted above.

The causes at lower stages of the model form the observations to the stages above. Hidden causes therefore provide a link between the stages. The states, in contrast, both connect the dynamics over time, to ignore temporal discontinuities [26], and mediate the effects of the causes on the stimuli [6]. In the feedback case, the model generates top-down predictions such that the neural activity at one stage predicts the activity at a lower stage. The predictions from a higher level are sent through feedback connections to be compared to the actual activity. This yields a model uncertainty error that is forwarded to subsequent stages to update the population activity and improve prediction. Such a top-down process repeats until the bottom-up stimuli transformation process no longer imparts any new information. That is, there are no unexpected changes in the stimuli that the model cannot predict. Once this occurs, if the model is able to synthesize the input stimuli accurately using the uncovered features, then it means that it has previously seen a similar observation [27].

In short, a predictive-coding model has two processing pathways. Recurrent, top-down connections carry predictions about the activity to the lower model levels. These predictions reflect past experience [6]. They form priors to disambiguate the incoming sensory inputs. Bottom-up connections relay prediction errors to higher levels to update the physical causes. The interaction of the feed-forward and feedback processes [28] on the causes enables robust object analysis [29] from the observed stimuli. A well-trained predictive-coding model should thus distinguish between objects, despite learning about them in an unsupervised way. This only occurs if the models preserve some notion of perceptual difference.

### III. DEEP PREDICTIVE CODING

We propose an efficient architecture for the above hierarchical, latent-variable model. This model is suitable for uncovering discriminative details from the stimuli.

More specifically, we consider a faster, convolutional DPCN, the ADPCN. ADPCNs can extract highly sparse, invariant features for either dynamic or static stimuli (see Section III-A). We then show how to effectively infer the ADPCN’s latent variables using a fast proximal gradient scheme (see Section III-B). This optimization process permits effectively forming deep feature hierarchies that preserve perceptual differences. It typically overcomes the learning stagnation that we observed in DPCNs. Convergence properties are presented in the online appendix (see Appendix A in the supplementary material). In this appendix, we prove why stagnation occurs in the DPCNs. We also quantify the convergence rate of the DPCNs and ADPCNs to show that the latter, theoretically, converge more quickly than the former. This justifies our new inference process, as do our empirical results.

### A. ADPCN Cost Functions

ADPCNs consist of two phases at each stage, which is outlined in Fig. 1. The first phase entails inferring the hidden states, which are a feature-based representation used to describe the stimuli. States are formed at the first phase via sparse coding in conjunction with a temporal-state-space model. Stimuli are mapped to an over-complete dictionary of convolutional filters. Subsequent ADPCN stages follow the same process, with the only change being that the hidden causes assume the role of the observed stimuli.

We define state inference via a least-absolute-shrinkage-and-selection-operator (LASSO) cost. We present this for the case of single-channel stimuli, for the ease of readability. The extension to multiple channels is straightforward.

---

For the above framework, both feed-forward, bottom-up, and feedback, top-down processes are used to characterize observed stimuli. For the feed-forward case, the observed stimuli are propagated through the model to extract progressively abstract details. The stimuli are first converted to a series of states that encode either spatial, temporal, or spatio-temporal relationships. The type of relationship depends on the stimuli being considered. These states are then made invariant to various transformations, thereby forming hidden causes. The causes are latent variables that describe the environment, as we noted above.

The causes at lower stages of the model form the observations to the stages above. Hidden causes therefore provide a link between the stages. The states, in contrast, both connect the dynamics over time, to ignore temporal discontinuities [26], and mediate the effects of the causes on the stimuli [6]. In the feedback case, the model generates top-down predictions such that the neural activity at one stage predicts the activity at a lower stage. The predictions from a higher level are sent through feedback connections to be compared to the actual activity. This yields a model uncertainty error that is forwarded to subsequent stages to update the population activity and improve prediction. Such a top-down process repeats until the bottom-up stimuli transformation process no longer imparts any new information. That is, there are no unexpected changes in the stimuli that the model cannot predict. Once this occurs, if the model is able to synthesize the input stimuli accurately using the uncovered features, then it means that it has previously seen a similar observation [27].

In short, a predictive-coding model has two processing pathways. Recurrent, top-down connections carry predictions about the activity to the lower model levels. These predictions reflect past experience [6]. They form priors to disambiguate the incoming sensory inputs. Bottom-up connections relay prediction errors to higher levels to update the physical causes. The interaction of the feed-forward and feedback processes [28] on the causes enables robust object analysis [29] from the observed stimuli. A well-trained predictive-coding model should thus distinguish between objects, despite learning about them in an unsupervised way. This only occurs if the models preserve some notion of perceptual difference.

### III. DEEP PREDICTIVE CODING

We propose an efficient architecture for the above hierarchical, latent-variable model. This model is suitable for uncovering discriminative details from the stimuli.

More specifically, we consider a faster, convolutional DPCN, the ADPCN. ADPCNs can extract highly sparse, invariant features for either dynamic or static stimuli (see Section III-A). We then show how to effectively infer the ADPCN’s latent variables using a fast proximal gradient scheme (see Section III-B). This optimization process permits effectively forming deep feature hierarchies that preserve perceptual differences. It typically overcomes the learning stagnation that we observed in DPCNs. Convergence properties are presented in the online appendix (see Appendix A in the supplementary material). In this appendix, we prove why stagnation occurs in the DPCNs. We also quantify the convergence rate of the DPCNs and ADPCNs to show that the latter, theoretically, converge more quickly than the former. This justifies our new inference process, as do our empirical results.

### A. ADPCN Cost Functions

ADPCNs consist of two phases at each stage, which is outlined in Fig. 1. The first phase entails inferring the hidden states, which are a feature-based representation used to describe the stimuli. States are formed at the first phase via sparse coding in conjunction with a temporal-state-space model. Stimuli are mapped to an over-complete dictionary of convolutional filters. Subsequent ADPCN stages follow the same process, with the only change being that the hidden causes assume the role of the observed stimuli.

We define state inference via a least-absolute-shrinkage-and-selection-operator (LASSO) cost. We present this for the case of single-channel stimuli, for the ease of readability. The extension to multiple channels is straightforward.
Definition 2 (State LASSO Cost): Let \( \gamma_{i,t} \in \mathbb{R}^k \) be the hidden states at time \( t \) and at model stage \( i \). Let \( C_i \in \mathbb{R}^{k \times k} \) be a hidden-state-transition matrix. Let \( D^+_i \in \mathbb{R}^{k \times k-1} \) be a Toeplitz-form matrix with \( q_i \in \mathbb{N}_+ \) filters structured as in [30]. The state-inference cost function to be minimized, with respect to \( \gamma_{i,t}, C_i \), and \( D^+_i \), is given by (1), shown at the bottom of the page, where \( \kappa_{0,t} = \gamma_t \). The first term in this cost quantifies the \( L_2 \) prediction error, \( \epsilon_{i,t} = \kappa_{i-1,t} - D^+_i \gamma_{i,t} \), at stage \( i \). The aim is to ensure that the local reconstruction error between stages is minimized. The second term constrains the next-state dynamics to be described by the state-transition matrix. For state stimuli, indexed by \( t \), the state feedback is replaced by \( \kappa_{i,t} = D^+_i \gamma_{i,t+1} \). The strength of the recurrent feedback is \( \lambda_{i,t} \). The strength of the recurrent feedback is \( \gamma_{i,t} \). The final term enforces \( L_1 \) sparsity of the states, \( \lambda_{i,t} \), sparse to make the state-space representation consistent. Without such a norm penalty, the innovations would not be sparse due to the feedback. The final term enforces \( L_1 \) sparsity of the states, with the amount controlled by \( \lambda_{i,t} \).

Proposition 1: Let \( \gamma_{i,t} \in \mathbb{R}^k \) be the hidden states at time \( t \) and at model stage \( i \). Let \( D^+_i \in \mathbb{R}^{k \times k-1} \) be a Toeplitz-form matrix of \( q_i \in \mathbb{N}_+ \) filters. The state-vector multiplication \( D^+_i \gamma_{i,t} \) is functionally equivalent to convolution.

Proposition 2: Let \( \gamma_{i,t} \in \mathbb{R}^k \) be the hidden states at time \( t \) and at layer \( i \). Let \( C_i \in \mathbb{R}^{k \times k} \) be a hidden-state-transition matrix. Let \( D^+_i \in \mathbb{R}^{k \times k-1} \) be a Toeplitz-form matrix with \( q_i \in \mathbb{N}_+ \) filters. For \( \alpha_i, \lambda_{i,t} \in \mathbb{R}_{+} \), the hidden state cost function \( L_1(\gamma_{i,t}, \kappa_{i,t}, C_i, D^+_i; \alpha_i, \lambda_{i,t}) \) is convex.

Note that the state LASSO may not be convex with respect to the parameters as the parameters are updated during inference. We may thus recover only local minimizers, not global ones.

The state-based feature representations constructed by the first phase are not guaranteed to be invariant to various transforms. Discrimination can be impeded, as a result. The second ADPCN processing phase thus entails explicitly imposing this behavior. Local translation invariance is attained by leveraging the spatial relationships of the states in neighborhoods via the max-pooling of states. Invariance to more complex transformations, like rotation and spatial frequency, is made possible through the inference of subsequent hidden causes.

Sparse cause inference is driven by a LASSO-based cost that captures nonlinear dependencies between components in the pooled states. We present this cost for the case of a single convolutional layer, for the ease of readability. We utilize multiple layers in our simulations.

Definition 3 (Cause LASSO Cost): Let \( \gamma_{i,t} \in \mathbb{R}^k \) be the hidden states and \( \kappa_{i,t} \in \mathbb{R}^d \) be the bottom-up hidden causes at time \( t \) and model stage \( i \). Let \( \kappa^s_{i,t} \in \mathbb{R}^d \) be the top-down inferred causes. Let \( C_i \in \mathbb{R}^{k \times k} \) be a hidden-state-transition matrix. Let \( D^+_i \in \mathbb{R}^{k \times k-1} \) be a Toeplitz-form matrix of \( q_i \in \mathbb{N}_+ \) filters. Let \( G_i \in \mathbb{R}^{d \times k} \) be an invariant Toeplitz matrix. The hidden-cause cost to be minimized, with respect to \( \kappa_{i,t} \) and \( G_i \), is given by (2), as shown at the bottom of the page, where we have that \( \lambda_{i,t,k'} = \alpha_t^i(1 + \exp(-[G_i \kappa_{i,t}^s]_{k'})) \). The first term in this cost models the multiplicative interaction of the causes \( \kappa_{i,t} \) with the sum-pooled states \( \gamma_{i,t} \) through an invariant matrix \( G_i \in \mathbb{R}^{d \times k} \). This characterizes the shape of the sparse prior on the states. That is, the invariant matrix is adapted such that each component of the causes is connected to element groups in the accumulated states that cooccur frequently. Cooccurring components typically share common statistical regularities, thereby yielding locally invariant representations [32]. The second term specifies that the difference between the bottom-up \( \kappa_{i,t} \) and top-down inferred causes \( \kappa^s_{i,t} \) should be small, with the term weight specified by \( \eta_{i,t} \). The final term imposes \( L_1 \) sparsity, with the amount controlled by \( \lambda_{i,t} \), to prevent the intermediate representations from being dense.

The causes obtained by solving the above LASSO cost will behave somewhat like complex cells in the visual cortex [33]. Similar results are found in temporally coherent networks [34], albeit without guaranteed feature invariance.

As with the state-inference cost, we employ \( L_1 \) sparsity in the hidden-cause cost for practical reasons, even though we would prefer \( L_0 \) sparsity for its theoretical appeal.

Proposition 3: Let \( \gamma_{i,t} \in \mathbb{R}^k \) be the hidden states and \( \kappa_{i,t} \in \mathbb{R}^d \) be the hidden causes at time \( t \) and model stage \( i \). Let \( C_i \in \mathbb{R}^{k \times k} \) be a hidden-state-transition matrix, \( D^+_i \in \mathbb{R}^{k \times k-1} \) be a Toeplitz-form matrix of filters, and \( G_i \in \mathbb{R}^{d \times k} \) be an invariant Toeplitz matrix. Let \( \alpha^i_t, \lambda_{i,t}^j, \eta_{i,t}^j \in \mathbb{R}_{+} \) and \( \lambda_{i,t} \in \mathbb{R}_+^k \). The hidden-cause cost-function \( L_2(\gamma_{i,t}, \kappa_{i,t}, G_i; \alpha^i_t, \lambda_{i,t}^j, \eta_{i,t}^j, \lambda_{i,t}) \) is convex for appropriate parameter values.

Practically, we have found that traditional \( L_0 \) sparsity in the ADPCN preempts learning. The causes are often too sparse to act as priors. Large errors continuously accumulate, so the ADPCNs are unable to reduce the residual prediction errors. Approximating the \( L_0 \) term is often a better option. We will explore it more in our future endeavors.

ADPCNs and the DPCNs our lab proposed [7], [8] have almost the same cost functions that we outline above. They both build unsupervised representations of input stimuli [35] via a free-energy principle [36]. The difference is that ADPCNs implement convolution by way of the Toeplitz-form matrix of filters. The original DPCNs rely on nonconvolutional filters. The expressive power of these DPCNs is thus quite

\[
L_1(\gamma_{i,t}, \kappa_{i,t}, C_i, D^+_i; \alpha_i, \lambda_{i,t}) = \frac{1}{2} \left( \| \kappa_{i-1,t} - D^+_i \gamma_{i,t} \|_2^2 + \alpha_i \| \gamma_{i,t} - C_i \gamma_{i-1,t} \|_1 + \sum_{k=1}^{k_i} [\lambda_{i,t}]_k \| \gamma_{i,t} \|_k \right)
\]

\[
L_2(\gamma_{i,t}, \kappa_{i,t}, G_i; \alpha^i_t, \lambda_{i,t}^j, \eta_{i,t}^j, \lambda_{i,t}) = \frac{1}{2} \left( \sum_{j=1}^{n} \sum_{k=1}^{k_i} \| \lambda_{i,t}^j \| \gamma_{i,t}^j \|_k + \eta_{i,t}^j \| \kappa_{i,t} - \kappa^s_{i,t} \|_2^2 + \lambda_{i,t} \| \kappa_{i,t} \|_1 \right)
\]
poor for complex stimuli compared to the ADPCNs. ADPCNs can take advantage of local spatial coherence in addition to temporal coherence. They, hence, require fewer filters than DPCNs to extract meaningful stimuli representations.

Defining states and causes as we have specified above has significant advantages. ADPCNs are, for instance, incredibly parameter efficient compared to standard recurrent-convolutional autoencoders. Few filters often are needed to take advantage of local spatial coherence in addition to temporal coherence. They, hence, require fewer filters than DPCNs to extract meaningful stimuli representations. ADPCNs can, indeed, be kernelized to realize nonlinear, nonparametric state-space updates, further increasing the ADPCNs’ expressive power without impacting inference times. The hidden states are clamped via a soft thresholding function implicit to the proximal operator, which leads to a sparse solution. The states are then spatially max pooled over local neighborhoods, using nonoverlapping windows, to reduce their resolution, γt+1 = POOL(γt+1).

Definition 5 (Feed-Forward, Bottom-Up Cause Inference): Let γt, ∈ Rk be the hidden states and κt, ∈ Rd be a hidden-state-transition matrix, D+ ∈ Rk×k−1 be a Toeplitz-form matrix of qi ∈ N filters. For an inertial sequence βm ∈ R+ and an adjustable step size τ+ ∈ R+, the hidden-state inference process, indexed by iteration m, is given by (3), as shown at the bottom of the page, with L1(κt, κt, Ci, D+ ; α, λ, ti, ti) = D+ (κti−1 − D+κt−1) + aλi(κt−1). Here, use a Nesterov smoothing, Ωi(κt−1) = maxΩi(κ−1)Ωi(κt−1)−Ci(κ−1)−1, to approximate the nonsmooth state transition. Small values for the hidden states are clamped via a soft thresholding function implicit to the proximal operator, which leads to a sparse solution. The states are then spatially max pooled over local neighborhoods, using nonoverlapping windows, to reduce their resolution, γt+1 = POOL(γt+1).

Definition 6 (Feed-Forward, Bottom-Up Cause Inference): Let γt, ∈ Rk be the hidden states and κt, ∈ Rd be a hidden-state-transition matrix, D+ ∈ Rk×k−1 be a Toeplitz-form matrix of qi ∈ N filters, and Gt ∈ Rd×kd be an invariant Toeplitz matrix. For an adjustable step size τ+ ∈ R+ and an inertial sequence βm ∈ R+ (3), as shown at the bottom of the page, the cause iterates (4), as shown at the bottom of the page, with L1(κt, κt, Ci, D+ ; α, λ, ti, ti) = −a(κt−1)+exp(−Gt(κt−1))γt+1+2η(κt−1)−κt−1. Small values for the hidden causes are clamped via an implicit soft thresholding function, leading to a sparse solution. The inferred causes are used to update the sparsity parameter λt+1 = a(1 + exp(−unpool(Gtκt+1))) via spatial max unpooling.

In both cases, the step size is bounded by the Lipschitz constant of the LASSO cost to be solved. The choice of the inertial sequence greatly affects the convergence properties of the optimization.

Proposition 4: Let γt, ∈ Rk be the hidden states and κt, ∈ Rd be the hidden causes. The state iterates {γi, κm}∞m=1 strongly converge to the global solution of L1(γi, κt, Ci, D+ ; α, λ, ti) for the accelerated proximal gradient scheme. Likewise, the cause iterates {κm}∞m=1 strongly converge to the global solution of L2(γi, κm, Gt ; α, λ, ti) at a subpolynomial rate. This occurs when using the inertial sequences βm, βm = (k−1)/m, where k depends polynomially on m.

In this bottom-up inference process, there is an implicit assumption that the top-down predictions of the causes are available. This, however, is not the case for each iteration of a minibatch being propagated through the ADPCN. We, therefore, consider an approximate, top-down prediction using the states from the previous time instance and, starting from the first stage, perform bottom-up inference using this prediction.

Definition 6 (Feedback, Top-Down Cause Inference): At the beginning of every time step t, using the state-space model
at each stage, the likely top-down causes, $\kappa’_{t-1,i} \in \mathbb{R}^{d_t-1}$, are predicted using the previous states $y_{t-1,i} \in \mathbb{R}^{k_t}$ and the causes $\kappa_{t,i} \in \mathbb{R}^d$. That is, for the filter dictionary matrix, the top-down update in (5), as shown at the bottom of the previous page, is performed, except for the last stage, wherein $\kappa’_{t+1,i} = \kappa_{t,i}$. This minimization problem has an algebraic expression for the global solution: $[y’_{t,i}]_k = [C_i y_{t-1,i}]_k$, whenever $\alpha’_{t,i} \lambda_{t,i} < \lambda_t$, and zero otherwise.

These top-down predictions serve an important role during inference, as they transfer abstract knowledge from higher stages into lower ones. The overall representation quality is thereby improved. The predictions also modulate the representations due to state zeroing by the sparsity hyperparameter.

Alongside the state and cause inference is a learning process for fitting the ADPCN parameters to the stimuli. Here, we consider gradient-descent training without top-down infor-
mation, which is performed once inference has stabilized for a given minibatch. An overview of this procedure is presented in Alg. 1.

The inference procedure in Alg. 1 is different from the one proposed in [7] and [8]. Our lab’s original DPCN relies on proximal gradients with an extra-gradient rule that is an almost-linear combination of the previous state and cause iterates. The ADPCNs leverage potentially nonlinear iterate combinations. This has the effect of taking larger steps along the error surface without diverging. In essence, the DPCN inference process is overly conservative in its updates. It also has issues that we outline in the online appendix (see Appendix A in the supplementary material).

Another change is that multiple iterations of top-down feedback are executed in the ADPCNs. As the recurrent processes unfold in time, the APDCN is used over and over to apply an increasing number of nonlinear transformations to the stimuli. This has the effect of simulating the propagation of the stimuli through an increasingly deeper, feed-forward network but without the overhead of adding and learning more network parameters [37]. This promotes the formation of more expressive states and causes that quickly converge as the recurrent processes proceed over time. It suggests that the ADPCNs have a stable, self-organizing mechanism that minimizes surprise well [38]. Our lab’s DPCNs [7], [8] only consider a single feedback iteration. A great many hierarchical stages are needed to replicate the above behavior. However, a lack of reasonable convergence prevents this from occurring. They hence, do not learn to preserve perceptual differences.

IV. Simulation Results

We now assess the capability of our inference strategy for unsupervised ADPCNs. We focus on static visual stimuli (see Section IV-A). We demonstrate that ADPCNs uncover meaningful feature representations. They do so more quickly than convolutional DPCNs that rely on a nonaccelerated, proximal-gradient update (see Section IV-B). We show that the improved inference offered by ADPCNs permits them to exceed the performance of deep unsupervised networks and behave similarly to deep supervised networks.

A. Simulation Preliminaries

1) Data and Preprocessing: We rely on five datasets for our simulations. Two of these, MNIST and FMNIST, contain single-channel, static visual stimuli. The remaining three, CIFAR-10, CIFAR-100, and STL-10, contain multichannel, static visual stimuli. We whiten each dataset and zero their means. We use the default training and test set definitions for each dataset.

2) Training and Inference Protocols: For learning the DPCN and ADPCN parameters, we rely on ADAM-based gradient descent with minibatches [39]. We set the initial learning rate to \( \eta_0 = 0.001 \), which helps prevent over-shooting the global optimum. The learning rate is decreased by half every epoch. We use exponential decay rates of 0.9 and 0.99 for the first- and second-order gradient moments in ADAM, respectively, which are employed to perform bias correction and adjust the per-parameter learning rates. An epsilon additive factor of \( 10^{-8} \) is used to preempt division by zero. We use an initial forgetting factor value of \( \theta_0 = 0.7 \). This factor is increased by a tenth every thousand minibatches to stabilize convergence. We consider a minibatch size of 32 randomly selected samples to ensure good solution quality [40].

For DPCN and ADPCN inference, we primarily set the sparsity parameters to \( \lambda_1, \lambda'_1 = 0.2, \lambda_2, \lambda'_2 = 0.25, \) and \( \lambda_3, \lambda'_3 = 0.35 \). Such values permit retaining much of the visual content in the first two stages while compressing it more pronouncedly in the third. Some stimuli datasets have slightly altered parameter values. Due to the static nature of the stimuli, we do not have temporal state feedback. We do, however, propagate the cause-state difference between stages. This is a slight modification to the network diagram shown in Fig. 1. We set the feedback strengths, for most simulations, to \( a_1, a_2 = 1 \) and \( a_3 = 3 \). In Fig. 1, these variables correspond to the strength of the temporal, intralayer feedback, but here they are used for nontemporal, interlayer state feedback. The stronger feedback amount in the third stage aids in suppressing noise without adversely impacting the earlier stages’ priors. We fix the causal sparsity constants to \( a_1, a_2, a_3 = 1 \). We terminate the accelerated and regular inference processes after 500 and 1000 iterations, respectively, per minibatch. A significantly lower number of iterations is used in the former case since the new inertial sequence facilitates quick convergence.

B. Simulation Results and Discussions

1) Network Architecture: As we note in the previous section, our lab’s DPCNs are nonconvolutional. For our simulations, we consider the same Toeplitz-based convolution as the ADPCNs. This is done to highlight that the improvement in feature quality occurs due to the faster inference strategy.

We consider the same architecture for the convolutional DPCNs and ADPCNs. At the first stage, we use 128 states with \( 5 \times 5 \) filters and 256 causes with \( 5 \times 5 \) filters. This yields an over-complete reconstruction basis. At the second and third stages, we use 128 states and 256 states with \( 5 \times 5 \) and \( 5 \times 5 \) filters, respectively. For these two stages, we use 512 causes and 1024 causes with \( 5 \times 5 \) and \( 5 \times 5 \) filters, respectively. We perform \( 2 \times 2 \) max pooling between the states and the causes at each stage.

Due to the choice of filter sizes, the ADPCNs will typically have a worse reconstruction error but better recognition rate in the later stages. Using larger filters in the early stages and smaller ones in the later stages permits implementing traditional predictive-coding behaviors. That is, the reconstruction error decreases deeper in the hierarchy, as parts of the stimuli are explained away in an iterative manner.

2) Simulation Results: Simulation findings are presented in Figs. 2 and 3 for the single-channel MNIST and FMNIST datasets, respectively. Findings for the multichannel CIFAR-10/100 and STL-10 datasets are, respectively, shown in Figs. 4 and 5. The results in these figures are for after two epochs. Some results are presented in Appendices B and C in the supplementary material.

For these datasets, the ADPCNs are successful in quickly uncovering invariant representations. Most of the columns in the invariance matrix group dictionary elements have very similar orientation and frequency while being insensitive to translation [see Figs. 2(a)–5(a)]. Likewise, for each active invariance-matrix column, a subset of the dictionary elements are grouped by orientation and spatial position, which indicates invariance to other properties like spatial frequency and center position [see Figs. 2(b)–5(b)]. The convolutional DPCNs, in comparison, have representations that are significantly altered by transformations other than translation. This
Fig. 2. Comparison of accelerated proximal gradient inference and learning (left, blue) and proximal gradient inference and learning (right, red) the MNIST dataset. The presented results are shown after training with minibatches for two epochs. (a) Polar scatter plots of the orientation angles versus spatial frequency for the first-layer causes. (b) Line plots of the normalized center positions with included orientations for the first-layer causes. For both (a) and (b), we fit Gabor filters to the first-layer causes; locally optimal filter parameters were selected via a gradient-descent scheme. The plots are color-coded according to the connection strength between the invariant matrix and the observation matrix in the first network layer. Higher connection strengths indicate subsets of dictionary elements from the observation matrix that are most likely active when a column of the invariant matrix is also active. If a DPCN has been trained well, then the filters should have a small orientation-angle spread. Each plot represents a randomly chosen column of the first-layer invariance matrix. (c)–(e) Back-projected causes from the first, second, and third layers of the networks, respectively. Each plot represents a randomly chosen column of the first-layer invariance matrix. (f)–(g) Reconstructed instances from a random batch at the first and third layer, respectively. For each layer, we also assess the feature similarity between denoted using progressively more vivid shades of either blue (accelerated proximal gradients) or red (proximal gradients). If a DPCN has been trained well, then there should be few to no duplicate filters. There, hence, should not be any conspicuous blocky structures along the main diagonal of the VAT similarity plots. (f)–(g) Reconstructed instances from a random batch at the first and third layer, respectively. For each layer, we also assess the feature similarity between the original training samples and the reconstructed versions and provide corresponding scatter plots. If a DPCN reconstructs the input samples well, then there should be a strong linear relationship between the features. Higher distributional spreads and shifts away from the main diagonal indicate larger reconstruction errors.

occurs because subsets of the dictionary elements are not grouped according to various characteristics. Discrimination performance, hence, can suffer for stimuli samples that are slightly altered.

Also, the ADPCNs learn meaningful filters from the stimuli. The first two stages of our ADPCNs have causal receptive fields that mimic the behavior of simple and complex cells in the primate vision system [see Figs. 2(c)–(d)–5(c)–(d)]. The fields for the first stage are predominantly divided into two types: low-frequency and high-frequency, localized bandpass filters. The former mainly encode regions of uniform intensity and color along with slowly varying texture. The latter describe contours and hence, sharp boundaries. Such filters permit accurately reconstructing the input stimuli [see Figs. 2(f)–5(f) and Fig. B1(a)–(e)]. The second-stage receptive fields are nonlinear combinations of those in the first that are activated by more complicated visual patterns, such as curves and junctions. A similar division of receptive fields into two categories is often encountered in the second ADPCN stage. More filters are activated by contours, however, than in the first stage. For both stages, the filters are mostly unique, which is captured in the ordered similarity plots [see Figs. 2(c)–(d)–5(c)–(d)]. Beyond two stages, the ADPCN receptive fields encompass entire objects [see Figs. 2(e)–5(e)]. They are, however, average representations, not highly specific ones, due to the limited number of causes [see Figs. 2(g)–5(g)]. The backgrounds in the visual stimuli are often suppressed at the third stage for CIFAR-10/100 and STL-10, which greatly enhance recognition performance [see Fig. B1(c)–(e)]. There are no backgrounds for MNIST and FMNIST, so recognition is predominantly driven by the whole-object receptive fields in the third stage [see Fig. B1(a)–(b)]. The ordered similarity plots indicate that none of the third-stage filters appear to be duplicated for either dataset. This trend also holds for the first- and second-stage receptive fields. This implies that the ADPCNs emphasize the extraction of nonredundant features to form a complete visual stimuli basis at each stage.
The filters learned by ADPCNs are not only sensitive to whole objects. They are also attuned to stylistic changes of objects. As shown in Fig. C1, the filter-derived cause features tend to segregate instances of objects that visually differ. This occurs even within a given object class.

Convolutional DPCNs, in contrast, do not stabilize to viable receptive fields at the same rate as the ADPCNs [see Figs. 2(c)–(d)–5(c)–(d)]. For MNIST, the first-stage DPCN receptive fields have some localized bandpass structure that is similar to Gabor filters. The overall spread of the fields makes it difficult to accurately detect abrupt transitions and hence, recreate the input stimuli, though. The reconstructions thus are heavily distorted and blurred [see Fig. 2(f) and Fig. B1(a)]. For FMNIST, the first-stage receptive fields focus on low-frequency details, such as either constant grayscale values or slow-changing grayscale gradients. They also focus on higher-frequency details, such as periodic texture. While some of the causes become specialized bandpass-like filters, there are not enough to adequately preserve sharp edges. The stimuli reconstructions are thus also distorted, which removes much of the high-frequency content [see Fig. 3(f) and Fig. B1(b)].

Similar results are encountered for CIFAR-10/100 and STL-10 [see Figs. 4(f)–5(f) and Fig. B1(c)–(e)]. For all of the datasets, the second-stage receptive fields become even less organized than in the first stage. They are mostly activated by blob-like visual patterns, which do not preserve enough visual content for recreating a close resemblance of the input stimuli beyond the first stage. The convolutional DPCNs are unable to learn relevant representations in the third stage, as a consequence. The receptive fields for this network stage are unique, by virtue of being essentially random. They are, however, largely useless in extracting stimuli-specific details [see Figs. 2(e)–5(e)]. This further degrades the reconstruction quality to where the inputs are unrecognizable [see Figs. 2(g)–5(g)]. Discrimination is adversely impacted due to this severe lack of identifying characteristics [see Fig. B1(a)–(e)]. The filter redundancy also impacts learning good model priors. There is typically not enough unique information to be back-propagated to earlier stages to inform the choice of better receptive fields.

The causes formed by the ADPCNs and DPCNs specify invariant features that can be employed for discrimination. In Fig. B2, we show that the stage-aggregated ADPCN features yield high-performing unsupervised classifiers. They achieve state-of-the-art unsupervised recognition rates for each dataset. These recognition rates also are on par with deep networks trained in a supervised fashion, despite having orders of magnitude fewer parameters. Although the causal features from all stages have a positive net contribution, those from the third stage contribute the most to recognition performance [see Fig. B1(a)–(e)]. The DPCNs exhibit poor performance, in comparison. Only the first-stage features aid classification [see Fig. B1(a)–(e)]. The remainder largely worsens the recognition capabilities. For both the DPCNs and ADPCNs, we rely on a seven-nearest neighbor classifier with an unsupervised-learned metric distance [41] to label the stimuli samples. As we illustrate in Fig. C1, the causal features require nonlinear decision boundaries to distinguish between classes well. This occurs even in high-dimensional spaces, as class overlap occurs. Such a property motivates the use of nearest-neighbor classifiers.
3) Simulation Discussions: Our simulations indicate that ADPCNs were more effective at uncovering highly discriminative feature representations of visual stimuli than the original DPCN inference strategy. A trait that contributed greatly to the ADPCNs’ success was its significantly improved search rate.

As noted in Appendix A in the supplementary material, proximal-gradient-type schemes can undergo four separate search phases, some of which have different local convergence rates. In one of the phases, the constant-step regime, the states and causes undergo rapid improvements. However, in two phases, the local convergence rate is slow whenever the largest eigenvalue of a certain recurrence matrix is less than the current inertial-sequence magnitude. For linear inertial sequences, like those found in the proximal-gradient-based DPCNs, this condition occurs early during the optimization process. That is, for such sequences, the growth is initially very rapid and follows a logarithmic rate. Within just a few iterations, the sequence magnitude exceeds the eigenvalue, which preempts the fast constant-step regime. The rate of convergence becomes worse than sublinear. A large number of search steps is thus needed to move toward the global solution. However, the search frequently terminates before this happens due to reaching the maximum number of proximal-gradient iterations. The search, alternatively, stops early due to a lack of progress across consecutive iterations. In either case, the states and causes do not adequately stabilize for a given minibatch. The poor state and cause representations, naturally, are integrated into the filter dictionary matrices and invariant matrices during the learning updates. This disrupts the priors in the early hierarchy for future stimuli. The convergence is further stymied by cost rippling. Proximal-gradient-based optimization does not behave like a pure descent method in two out of the four phases. Such behavior is caused by the eigenvalues of another recurrence matrix being a pair of complex conjugates, which necessitates oscillating between the two. All of these factors make it difficult to propagate meaningful bottom-up information [42] beyond the first stage. The top-down details from higher stages are thus ineffective at modifying the priors to disambiguate stimuli [43].

The ADPCNs largely avoid these issues. After the constant-step regime, the search switches to one of two potentially slower phases. However, the APDCNs’ inertial-sequence growth rate is rather muted, as opposed to that of the DPCNs. The chance of exceeding the largest eigenvalue of the augmented, auxiliary-variable recurrence matrix is low for the ADPCNs. The searches thus can proceed unhindered toward the global solution. Moreover, since the eigenvalue-magnitude threshold is often not reached, the largest eigenvalue of the augmented, mapping recurrence matrix is real-valued, not complex. This means that the accelerated proximal gradients behave like a descent method. It thus will not experience localized cost rippling due to alternating between conjugate pairs. Both properties promote rapid stabilization of the states and causes, which expedite the formation of beneficial priors throughout much of the early-stage network inference. These priors facilitate the construction of transformation-insensitive feature abstractions in deeper network stages, due to the bottom-up forwarding of stimulus-relevant signals. The recurrent, top-down connections, in turn,
are also important aspects that aid in the ADPCNs’ success. The uniqueness of the bandpass filters is beneficial, as it creation of transformation-invariant features. These features are hence, ignored. This behavior biases in favor of valuable that do not contribute greatly to the reconstruction quality suitably alter the representation sparsity. Extraneous details emerge due to feedback from high-level visual network areas, which quickly reduce activity in lower areas to simplify the description of the stimuli to some of its most basic elements [50]. In doing so, alternate explanations are suppressed, and only the most dominant, fundamental causes of the stimuli remain [3]. These causes are oriented luminance contours. This rapid stabilization of filters is exactly functionality that we expect to encounter in an efficient predictive coding process. It is hence, well aligned with neurophysiological theory [51].

There are other traits that contribute to the success of the ADPCNs. For instance, the ADPCN’s first-stage receptive fields are largely similar to that of simple cells in the primary visual cortex. Simple cells often implement Gabor-like filters with generic preferred stimuli that correspond to oriented edges [48], [49]. Such filters are highly useful within the ADPCN. Relations between activations for specific spatial locations tend to be distinctive between objects in visual stimuli. Activations are also obtained, in a Gabor space, that facilitate the construction of naturally sparse representations. These representations can then be hierarchically extended, which is what we ultimately sought in the ADPCNs. Changes in object location, scale, and orientation can be reliably detected, within this Gabor space, thereby aiding in the creation of transformation-invariant features. These features permit near-perfect stimuli reconstructions at the first stage. The uniqueness of the bandpass filters is beneficial, as it permits the ADPCNs to fixate on nonredundant stimuli characteristics. How these filters arise and the rate at which they form are also important aspects that aid in the ADPCNs’ success.

They emerge due to feedback from high-level visual network areas, which quickly reduce activity in lower areas to simplify the description of the stimuli to some of its most basic elements [50]. In doing so, alternate explanations are suppressed, and only the most dominant, fundamental causes of the stimuli remain [3]. These causes are oriented luminance contours. This rapid stabilization of filters is exactly functionality that we expect to encounter in an efficient predictive coding process. It is hence, well aligned with neurophysiological theory [51].

The plots of center position and orientation angle versus center frequency for the ADPCNs indicate that the learned Gabor filters form a meaningful reconstruction basis for the visual stimuli. The entire center-position plane is thoroughly filled. There, hence, are filter impulse responses everywhere in space when a sufficient number of first-stage filters are learned. Likewise, the Gabor orientation range is well covered, implying that the filters can recognize edges at different angles. Taken together, both properties indicate that these first-stage filters can reconstruct the stimuli well. Given the similarities of the Gabor filters across the different stimuli datasets, it would appear that the inferred basis can recreate any natural-image stimuli well. The low reconstruction errors for the first stage of the ADPCN corroborate this claim. For the DPCNs, we also have that the center-position plane is well covered. However, the filters have poor spatial frequency bandwidths, so they are not activated much by contours. Only the most dominant edges in the visual stimuli will be well preserved. These usually are the object-background boundaries. Interobject details are not retained. While the first-stage DPCN filters also form a basis, it is a poor one for reconstructing natural-image.
stimuli. Without a meaningful, stable basis, the remainder of the network cannot build on it to reliably describe aspects of the stimuli. The slow convergence of proximal gradients prevents this, as we note above.

Beyond the first stage, the ADPCNs exhibit an increase in specificity, abstraction, and invariance for deeper hierarchies [52]. This behavior aligns well with the current understanding of the vision system [53]. It also aids in the ADPCNs’ success. The second-stage receptive fields, in the case of MNIST, become sensitive to curved sub-strokes for the handwritten digits, which is similar to prestriate cortex functionality [54]. For FMNIST, they emphasize abrupt transitions written digits, which is similar to prestriate cortex functionality [54].

For STL-10, the receptive fields are often elongated Gabor-like filters, which can be found in the prestriate cortex [57]. All of these features systematically focus on object-relevant visual details, thereby aiding recognition. At the deepest stage, the receptive fields are entirely object-specific, which is functionality somewhat akin to the neurons in the primate infero-temporal cortex [58]. It is also related to memory activity [59], [60]. The representations are additionally translation and rotation insensitive, similar to inferotemporal cortex neurons [61]–[63], and change little with respect to scale and spatial-frequency changes, similar to neurons in the middle temporal area [64], [65]. To our knowledge, such invariant, whole-object sensitivity within a single stage has not been witnessed in any existing predictive-coding model. Based on contemporary theories [66], we believe that the receptive-field feedback from the final stage contributes to the effective connectivity in the earlier network stages [67]. That is, the role of this final stage is to disambiguate local image components. This is done by creating a template that is fed back, which then selectively enhances object components and suppresses interfering background components [68]. This biologically plausible behavior is crucial for achieving high discrimination rates on CIFAR-10/100 and STL-10. For these datasets, the objects of interest are scattered amongst distractors.

As we show in our simulation results, the third stage of the ADPCN contributes more to recognition than the first two stages. Such a finding is well-aligned with previous studies. Although simple objects can be discriminated based on Gabor-like filters, complex objects require significant nonlinearities [69]. The later stages provide this functionality due to the conversion from states to pooled, sparse causes. The processes of pooling and sparsifying are highly nonlinear. They additionally lead to the emergence of complex-cell properties along with shift invariance [70] and spatial phase invariance [71].

Including Toeplitz-based convolution with spatial pooling has a major impact on performance. Spatial coherence behaviors that convolutional layers offer are essential for describing interobject relationships in complex visual stimuli. Pooling introduces invariance. Our promising recognition results in Appendix B in the supplementary material are a testament to this. The corresponding cause embeddings are too. They indicate that the ADPCNs self-organize the states and causes in an object-sensitive way, despite the lack of supervision. DPCNs struggle to do this, since they rely on nonconvolutional filters [7], [8]. Their cause embeddings often group the stimuli in a worse way than embeddings of the input stimuli. That is, their transformation of the stimuli often destroys much of the visual content. Even the convolutional extension to DPCNs struggle due to a lack of a good reconstruction basis. This shows that convolution alone is not sufficient to uncover good features. The networks must be paired with an efficient inference process. Inference for the convolutional and non-convolutional DPCNs is based on a slow, proximal-gradient-based approach. The later-stage cause embeddings for these networks are thus quite poor since the first- and second-stage bases do not stabilize quickly to meaningful filters. It is likely that DPCNs would benefit from a greedy, stage-wise training to initialize the networks in a meaningful way. This may overcome the poor convergence, to some extent. Our ADPCNs would likely benefit from it too. As well, the ADPCNs would probably benefit from fixing the first-stage states and causes. As we have shown in our simulations, these features do not change much across color-image datasets. Effort could be better spent on learning more advanced features in the later network stages.

Another benefit of including convolutional filters in the ADPCNs is that they help encode perceptual stimuli similarity and differences. By this, we mean that well-trained ADPCNs group related stimuli together and force unrelated stimuli apart in the causal feature space. Such behavior stems from learning visual, appearance-based features at local and global object scales. It also stems from iteratively sparsifying those features to selectively remove redundant details. Interlayer and intralayer feedback contribute too. Other network architectures, like convolutional autoencoders, should also be capable of accounting for perceptual differences. They, however, appear to currently lack both appropriate losses and regularizers, along with fundamental layer behaviors, to do this as effectively as ADPCNs.

V. Conclusion

Here, we have revisited the problem of unsupervised predictive coding. We have considered a hierarchical, generative network, the ADPCN, for reconstructing observed stimuli. This network is composed of temporal sparse-coding models with bidirectional connectivity. The interaction of the information passed by top-down and bottom-up connections across the models permits the extraction of invariant, increasingly abstract feature representations. Such representations preserve perceptual differences.

Our contribution in this article is a new means of inferring the underlying components of the feature representations, which are the sparse states and causes. Previously, a proximal-gradient-type approach has been used for this purpose. Despite its promising theoretical guarantees, though, it exhibits poor empirical performance. This practically limits the number of model stages that can be considered in DPCNs. It also extensively curtails the quality of the stimuli features that can be extracted. Here, we have considered a parallelizable, vastly accelerated proximal-gradient strategy that overcomes these issues. It allows us to go beyond the existing two-stage limitation, facilitating the construction of arbitrary-staged DPCNs. Each stage leads to increasingly enhanced performance for object analysis. The information from higher stages is propagated to earlier ones to form more effective stimuli priors for bottom-up processing. Most crucially, our optimization strategy immensely streamlined inference. Often, only one or two presentations of the stimuli are necessary to reach a stable filter dictionary and hence, a corresponding set of sparse states and sparse causes. Good object analysis
performance is hence, observed. The previous optimization approach requires many times more presentations before a stable set of filters is uncovered. The resulting features are not nearly as discriminative as those from our proposed strategy. We have mathematically proved that this stems from poor convergence-rate properties of the original inertial sequence used in DPCNs.

We have applied ADPCNs to static-image datasets. For MNIST and FMNIST, the ADPCNs learn initial-stage receptive fields that mimic aspects of the early stages of the primate vision system. The later network stages implement receptive fields that encompass entire objects. In the case of MNIST, the back-projected filters become pseudo averages of the hand-written numerical digits. Predominant writing styles are modeled well. For FMNIST, the back-projected filters are the various types of clothing and personal articles. General styles and some nuances are captured. To our knowledge, this is the first time that such object-scale receptive fields have been learned for predictive coding. This behavior helps yield unsupervised classifiers that achieve state-of-the-art performance. Such classifiers also outperform supervised-trained deep networks, which lends credence to the complicated feature inference and invariance process that we employ. It also supports the notion that the ADPCNs preserve perceptual differences. Similar results are witnessed for natural-image datasets, such as CIFAR-10/100. The later-stage receptive fields again encompass entire object categories. This yields promising features that achieve generalization rates almost equivalent to that of supervised-trained deep networks. This is despite our use of simple nearest-neighbor classifiers. It is also despite the fact that our ADPCNs have many times fewer convolutional filters than these other deep networks.

ADPCNs are readily applicable to spatio-temporal stimuli processing, much like the original DPCNs. This is because the ADPCNs implement a recurrent state-space model. For such modalities, like video, the ADPCN recognition performance maybe even better than for static stimuli, like images. This is because top-down, feedback connections impose temporal constraints on the learning process. We will investigate this in our future work from the context of solving time-varying problems [72]–[74]. We will also demonstrate their superiority compared to purely feed-forward convolutional-recurrent autoencoders. We will show that the lack of a top-down pathway impedes learning meaningful representations.

ADPCNs can also form the backbone of a general, unsupervised framework for stimuli learning and high-level understanding. In our future work, we will illustrate that architectural extensions of it are well suited for making extrapolations about environment dynamics. This has implications for many applications, including self-driving cars. In particular, we will show that these modified ADPCNs learn temporal and perceptual cues that permit predicting egocentric and allocentric events in the short-term future. For the egocentric events, the ADPCNs understand well that a moving camera will cause nonlinear transformations of visual characteristics across video frames. The ADPCNs thus can thus reliably predict, from a short history of video frames, the appearance of nonmobile objects and their locations in subsequent frames. This will be useful for estimating vehicle position and angle relative to objects of interest, which can help refine steering control inputs for close-quarters maneuvers. For the allocentric events, the ADPCNs understand well the dynamics of mobile objects for either a fixed or moving camera. The ADPCNs can hence, similarly predict the appearance and location of those objects. This will prove invaluable for gauging how pedestrians may react and hence, avoid collisions. We will show that both types of event predictions emerge due to the ADPCNs learning at multiple time scales across different stages. Multitime-scale prediction is something that existing recurrent models struggle to do well.

REFERENCES

[1] R. P. Rao and D. H. Ballard, “Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects,” Nature Neurosci., vol. 2, no. 1, pp. 79–87, 1999, doi: 10.1038/4580.

[2] K. Friston and S. Kiebel, “Predictive coding under the free-energy principle,” Phil. Trans. Roy. Soc. B, Biol. Sci., vol. 364, no. 1521, pp. 1211–1221, May 2009, doi: 10.1098/rspb.2008.0500.

[3] M. W. Spratling, “Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function,” Neural Comput., vol. 24, no. 1, pp. 60–103, Jan. 2012, doi: 10.1162/NECO_a_00222.

[4] M. V. Srinivasan, S. B. Laughlin, and A. Dubs, “Predictive coding: A fresh view of inhibition in the retina,” Proc. Roy. Soc. B, Biol. Sci., vol. 256, no. 1058, pp. 271–284, 1992, doi: 10.1098/rspb.1992.0085.

[5] J. F. M. Jehee, C. Rothkopf, J. M. Beck, and D. H. Ballard, “Learning receptive fields using predictive feedback,” J. Physiol., vol. 100, nos. 1–3, pp. 125–132, Jul. 2006, doi: 10.1016/j.jphysparis.2006.09.011.

[6] K. Friston, “Hierarchical models in the brain,” PLoS Comput. Biol., vol. 4, no. 11, Nov. 2008, Art. no. e1000211, doi: 10.1371/journal.pcbi.1000211.

[7] R. Chalasani and J. C. Principe, “Deep predictive coding networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), Scottsdale, AZ, USA, May 2013, pp. 1–13. [Online]. Available: http://arxiv.org/abs/1301.3541

[8] J. C. Principe and R. Chalasani, “Cognitive architectures for sensory processing,” Proc. IEEE, vol. 102, no. 4, pp. 514–525, Apr. 2014, doi: 10.1109/JPROC.2014.2307023.

[9] R. Chalasani and J. C. Principe, “Context dependent encoding using convolutional dynamic networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 9, pp. 1992–2004, Sep. 2015, doi: 10.1109/TNNLS.2014.2360060.

[10] W. Lotter, G. Kreiman, and D. Cox, “Deep predictive coding networks for video prediction and unsupervised learning,” in Proc. Int. Conf. Learn. Represent. (ICLR), Toulon, France, Apr. 2017, pp. 1–13. [Online]. Available: http://arxiv.org/abs/1605.08104

[11] K. Han, H. Wen, Y. Zhang, D. Fu, E. Culurciello, and Z. Liu, “Deep predictive coding network with locally recurrent processing for object recognition,” in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2018, pp. 9201–9213.

[12] R. Baker, M. Dexter, T. E. Hardwicke, A. Goldstone, and Z. Kourtzi, “Learning to predict: Exposure to temporal sequences facilitates prediction of future events,” Vis. Res., vol. 99, pp. 124–133, Jun. 2014, doi: 10.1016/j.visres.2013.10.017.

[13] P. Perruchet and S. Pacton, “Implicit learning and statistical learning: One phenomenon, two approaches,” Trends Cognit. Sci., vol. 10, no. 5, pp. 233–238, May 2006, doi: 10.1016/j.tics.2006.03.006.

[14] R. N. Aslin and E. L. Newport, “Statistical learning: From acquiring specific items to forming general rules,” Current Directions Psychol. Sci., vol. 21, no. 3, pp. 170–176, Jun. 2012, doi: 10.1177/0963721412436806.

[15] T. F. Brady and M. M. Chun, “Spatial constraints on learning in visual search: Modeling contextual cuing,” J. Exp. Psychol., Hum. Perception Perform., vol. 33, no. 4, pp. 798–815, 2007, doi: 10.1037/0096-1523.33.4.798.

[16] I. Zylberberg, J. T. Murphy, and M. R. DeWeese, “A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields,” PLoS Comput. Biol., vol. 7, no. 10, Oct. 2011, Art. no. e1002250, doi: 10.1371/journal.pcbi.1002250.

[17] N. L. Carlson, V. L. Ming, and M. R. DeWeese, “Sparse codes for speech predict spectrotemporal receptive fields in the inferior colliculus,” PLoS Comput. Biol., vol. 8, no. 7, Jul. 2012, Art. no. e1002504, doi: 10.1371/journal.pcbi.1002594.
Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
[61] E. T. Rolls, “Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas,” *Phil. Trans. Roy. Soc. London*, vol. 335, no. 1273, pp. 11–20, 1992, doi: 10.1098/rstb.1992.0002.

[62] M. J. Tovee, E. T. Rolls, and P. Azzopardi, “Translation invariance in the responses to faces of single neurons in the temporal visual cortical areas of the alert macaque,” *J. Neurophysiol.*, vol. 72, no. 3, pp. 1049–1060, Sep. 1994, doi: 10.1152/jn.1994.72.3.1049.

[63] E. Salinas and L. F. Abbott, “Invariant visual responses from attentional gain fields,” *J. Neurophysiol.*, vol. 77, no. 6, pp. 3267–3272, Jun. 1997, doi: 10.1152/jn.1997.77.6.3267.

[64] E. T. Rolls, G. C. Baylis, and C. M. Leonard, “Role of low and high spatial frequencies in the face-selective responses of neurons in the cortex in the superior temporal sulcus in the monkey,” *Vis. Res.*, vol. 25, no. 8, pp. 1021–1035, 1985, doi: 10.1016/0042-6989(85)90091-4.

[65] L. L. Lui, J. A. Bourne, and M. G. P. Rosa, “Spatial and temporal frequency selectivity of neurons in the middle temporal visual area of new world monkeys (*Callithrix jacchus)*,” *Eur. J. Neurosci.*, vol. 25, no. 6, pp. 1780–1792, Apr. 2007, doi: 10.1111/j.1460-9568.2007.05453.x.

[66] H. Liang, X. Gong, M. Chen, Y. Yan, W. Li, and C. D. Gilbert, “Interactions between feedback and lateral connections in the primary visual cortex,” *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 32, pp. 8637–8642, 2017, doi: 10.1073/pnas.1706183114.

[67] N. Ramalingam, J. N. J. Mcmanus, W. Li, and C. D. Gilbert, “Top-down modulation of lateral interactions in visual cortex,” *J. Neurosci.*, vol. 33, no. 5, pp. 1773–1789, Jan. 2013, doi: 10.1523/jneurosci.3825-12.2013.

[68] B. Epsstein, I. LilShitz, and S. Ullman, “Image interpretation by a single bottom-up top-down cycle,” *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 38, pp. 14298–14303, 2008, doi: 10.1073/pnas.0800968105.

[69] J. V. Shi, J. Wieland, R. T. Smith, and P. Sajda, “Perceptual decision making ‘through the eyes’ of a large-scale neural model of V1,” *Frontiers Psychol.*, vol. 4, p. 161, Apr. 2013, doi: 10.3389/fpsyg.2013.00161.

[70] D. H. Habel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *J. Physiol.*, vol. 160, no. 1, pp. 106–154, Jan. 1962, doi: 10.1113/jphysiol.1962.sp006837.

[71] Y. Lian, A. Almasi, D. B. Grayden, T. Kameneva, A. N. Burkitt, and H. Meffin, “Learning receptive field properties of complex cells in V1,” *PLOS Comput. Biol.*, vol. 17, no. 3, Mar. 2021, Art. no. e1007957, doi: 10.1371/journal.pcbi.1007957.

[72] Z. Zhang, L. Zheng, J. Weng, Y. Mao, W. Lu, and L. Xiao, “A new varying-parameter recurrent neural-network for online solution of time-varying Sylvester equation,” *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3135–3148, Nov. 2018, doi: 10.1109/TCYB.2017.2760883.

[73] Z. Zhang, Y. Lu, L. Zheng, S. Li, Z. Yu, and Y. Li, “A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality,” *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4110–4125, Dec. 2018, doi: 10.1109/TAC.2018.2810039.

[74] Z. Zhang, L. Zheng, T. Qiu, and F. Deng, “Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations,” *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 874–881, Feb. 2020, doi: 10.1109/TAC.2019.2921681.