Stable and Interpretable Unrolled Dictionary Learning

Bahareh Tolooshams
Demba Ba
School of Engineering and Applied Sciences
Harvard University

Reviewed on OpenReview: https://openreview.net/forum?id=e3S0Bl2RO8

Abstract

The dictionary learning problem, representing data as a combination of a few atoms, has long stood as a popular method for learning representations in statistics and signal processing. The most popular dictionary learning algorithm alternates between sparse coding and dictionary update steps, and a rich literature has studied its theoretical convergence. The success of dictionary learning relies on access to a “good” initial estimate of the dictionary and the ability of the sparse coding step to provide an unbiased estimate of the code. The growing popularity of unrolled sparse coding networks has led to the empirical finding that backpropagation through such networks performs dictionary learning. We offer the theoretical analysis of these empirical results through PUDLE, a Provable Unrolled Dictionary LEarning method. We provide conditions on the network initialization and data distribution sufficient to recover and preserve the support of the latent code. Additionally, we address two challenges; first, the vanilla unrolled sparse coding computes a biased code estimate, and second, gradients during backpropagated learning can become unstable. We show approaches to reduce the bias of the code estimate in the forward pass, and that of the dictionary estimate in the backward pass. We propose strategies to resolve the learning instability by tuning network parameters and modifying the loss function. Overall, we highlight the impact of loss, unrolling, and backpropagation on convergence. We complement our findings through synthetic and image denoising experiments. Finally, we demonstrate PUDLE’s interpretability, a driving factor in designing deep networks based on iterative optimizations, by building a mathematical relation between network weights, its output, and the training set.

1 Introduction

This paper considers the dictionary learning problem, namely representing data $x \in X \subset \mathbb{R}^m$ as linear combinations of a few atoms from a dictionary $D \in D \subset \mathbb{R}^{m \times p}$. Given $x$ and $D$, the problem of recovering the sparse (few non-zero elements) coefficients $z \in \mathbb{R}^p$ is referred to as sparse coding, and can be solved through the lasso (Tibshirani, 1996) (also known as basis pursuit (Chen et al., 2001)):

$$\ell_x(D) := \min_{z \in \mathbb{R}^p} \mathcal{L}_x(z, D) + h(z)$$

where $\mathcal{L}_x(z, D) = \frac{1}{2} \|x - Dz\|_2^2$, and $h(z) = \lambda \|z\|_1$. Specifically, the problem aims to recover a dictionary $D^*$ that generates the data, i.e.,

$$x = D^*z^*$$

where $z^*$ is sparse. Olshausen and Field (Olshausen & Field, 1997) introduced (2) in computational neuroscience as a model for how early layers of the visual cortex process natural images. Sparse coding has been widely studied and utilized in the statistics (Hastie et al., 2015) and signal processing communities (Elad, 2010). A few practical examples are denoising (Elad & Aharon, 2006), super-resolution (Yang et al., 2010), text processing (Jenatton et al., 2011), and classification (Mairal et al., 2009b), where it enables the extraction

1Source code is available at https://github.com/btolooshams/stable-interpretable-unrolled-dl
of sparse high-dimensional features representing data. Moreover, sparse modelling is ubiquitous in many other fields such as seismic signal processing (Nose-Filho et al., 2018), radar sensing for target detections (Bajwa et al., 2011), and astrophysics for image reconstruction from interferometric data (Akiyama et al., 2017). Furthermore, Cleary et al. (2017; 2021) use this model to learn a dictionary consisting of gene modules for efficient imaging transcriptomics.

Sparse coding has been utilized to construct neural architectures through approaches such as sparse energy-based models (Ranzato et al., 2007; 2008) or recurrent sparsifying encoders (Gregor & LeCun, 2010). The latter has initiated a growing literature on constructing interpretable deep networks based on an approach referred to as algorithm unrolling (Hershey et al., 2014; Monga et al., 2019). Deep unrolled neural networks have gained popularity as inference maps in recent years due to their computational efficiency and their performance in various domains such as image denoising (Simon & Elad, 2019; Tolooshams et al., 2021a; 2020), super-resolution (Wang et al., 2015), medical imaging (Solomon et al., 2020), deblurring (Schuler et al., 2016; Li et al., 2020), radar sensing (Tolooshams et al., 2021b), and speech processing (Hershey et al., 2014).

Prior to the advent of unrolled networks, gradient-based dictionary learning relied on analytic gradients computed from the lasso given the sparse code. With unrolled networks, automatic differentiation (Baydin et al., 2018), referred to as backpropagation (LeCun et al., 2012) in the reverse-mode, gained attention for parameter estimation (Tolooshams et al., 2018). The automatic gradient is obtained by backpropagation through the algorithm used to estimate the code. Automatic differentiation in reverse and forward-mode (Franceschi et al., 2017) is used in other areas, e.g., hyperparameter selection (Feurer & Hutter, 2019), and in a more relevant context, in the seminal work of LISTA (Gregor & LeCun, 2010). Other works demonstrated empirically the convergence of $\ell_1$-based dictionary learning by backpropagation through unrolled networks (Tolooshams et al., 2021a). Given finite computational power, Tolooshams et al. (2021a) convert sparse coding into an encoder by unrolling $T$ iterations of ISTA (Daubechies et al., 2004; Blumensath & Davies, 2008), and attach to it a linear decoder for reconstructing. Unrolled networks obtained in this manner suffer from two important limitations.

First, the sparse coding step in the forward pass computes a biased estimate of the code. This results, in turn, in a biased estimate of the backward gradient and, hence, a degradation of dictionary recovery performance. Second, as studied recently (Malézieux et al., 2022), inaccuracies in the early iterations of the unrolled network make backpropagation unstable. We address both of these shortcomings in this paper. Moreover, while Malézieux et al. (2022) analyze the gradient computed by backpropagation through unrolled sparse-coding networks, there is no known theoretical analysis of how weight updates using this gradient impact the recovery of a ground-truth code $z^*$, nor of their convergence to a ground-truth dictionary $D^*$.

This paper proposes a Provable Unrolled Dictionary LEarning (PUDLE) (Figure 1). We aim to recover $D^*$ by training the network using backpropagation with a learning rate of $\eta$. Three different choices affect the gradient: the number of unrolled iterations, the loss, and whether one backpropagates through the decoder only or through both the encoder and decoder. We highlight the impact of such choices on the convergence of the training algorithm. Backpropagation through the decoder results in the analytic gradient $g^{\text{dec}}_t$ using the code estimate $z_t$. The gradients $g^{\text{re-lasso}}_t$ and $g^{\text{re-ls}}_t$ are computed by backpropagation through the autoencoder using the lasso and least-squares objectives, respectively (Algorithm 2). We compare the gradients with the classical gradient-based alternating-minimization algorithm for dictionary learning (Chatterji & Bartlett, 2017) (i.e., cycling between sparse coding and dictionary update steps using the analytic gradient $\hat{g}$ (Algorithm 1)).
and provide a theoretical analysis of gradient-based recovery of the dictionary $D^*$. We provide sufficient conditions under which the gradient computation, hence the learning, is stable. Additionally, we show how using the reconstruction loss with backpropagation not only does not suffer from backpropagated instability but also ameliorates the propagation of the forward pass bias into the gradient estimate from the backward pass. Finally, we demonstrate the interpretability of the unrolled network. Our contributions are:

- **Unrolled sparse coding** Unlike prior work (Malézieux et al., 2022) that studies the ability of sparse coding to recover the solution of the lasso (1) given the current estimate of the dictionary (we call this local estimation), we study unrolled sparse coding for recovery of the true generating code in (2) (we call this global estimation). We provide sufficient conditions on the network and data distributions such that the forward pass recovers (Theorem 4.1) and preserves (Theorem 4.2) the correct code support. Assuming support identification, we show the linear convergence of the code estimated through the unrolled iterations to the solution of the lasso (Theorem 4.3). We provide an explicit code expression at unrolled layer $t$ and its error with respect the ground-truth code $z^*$; we highlight the biased estimate of the code when the forward pass strictly solves lasso (Theorems 4.4 and 4.5). Moreover, in a more general scenario, we show that the error in the code estimate is upper bounded by two terms, i.e., one associated with the dictionary error and the other to the bias of the estimate of code amplitude, due to $\ell_1$-based optimization (Theorem 4.6). The latter highlights that vanilla lasso ($\ell_1$-based) sparse coding computes a biased estimate of codes, and below we discuss strategies to either alleviate this bias in the forward pass or mitigate its propagation into the backward pass for dictionary learning.

- **Mitigation of coding bias propagation into dictionary learning** We study gradient estimation for dictionary learning in PUDLE. We decompose the upper bound on the gradient errors compared to the gradient direction to recover $D^*$ into terms involving the current dictionary error, the bias of the code estimate, and the lasso loss used to compute the gradient. We show that using only the reconstruction loss while backpropagating (i.e., $g^{ae}$-ls) results in the vanishing of the upper bound due to the usage of lasso loss. This means that given fixed $\lambda$, $g^{ae}$-ls ameliorates the propagation of the forward pass bias into the backward pass. Specifically, we show that $g^{ae}$-ls is a better estimator of the direction to recover $D^*$ than $g^{ls}$ and $g^{ae}$-lasso. Hence, weight updates using $g^{ae}$-ls converges to a closer neighbourhood of $D^*$ (Theorem 4.10). In a supervised image denoising task, we show that the advantage of $g^{ae}$-ls goes beyond dictionary learning; $g^{ae}$-ls results in better image denoising compared to $g^{dec}$. Furthermore, our network outperforms the sparse coding scheme in NOODL, a state-of-the-art online dictionary learning algorithm (Rambhatla et al., 2018) (Table 1). Moreover, we show that the bias in the estimate of $D^*$ vanishes as $\lambda_t = \lambda \nu^t$ (with $0 < \nu < 1$) decays within the forward unrolled layers (Figure 16). This strategy, supported by Theorem 4.4, results in an unbiased estimate of the code $z^*$ and recovery of $D^*$ (Theorem 4.11).

- **Stability of unrolled learning** Our approach to resolve the instability issue of backpropagation in unrolled networks is two-fold. First, we show that under proper dictionary initialization, the instability of the gradient $g^{ae}$-lasso computation, studied by Malézieux et al. (2022), as $T$ increases is resolved. We give a condition under which the code support is identified and recovered after one iteration and, hence, gradient computation stays stable. Second, in the absence of support identification in early iterations, we propose to use the gradient $g^{ae}$-ls which resolves the stability issue introduced by lasso loss in the backward pass. We highlight this stability through image denoising training without gradient explosion (Figure 6).

- **Interpretable sparse codes and dictionary** Prior work has discussed algorithm unrolling for designing interpretable deep architectures based on optimization models (Monga et al., 2019), or interpretability of sparse representations in dictionary learning models (Kim et al., 2010). However, there is no known work to mathematically characterize the interpretability of unrolled network architectures. In this regard, first, we construct a mathematical relation between learned weights (dictionary) at gradient convergence and the training data (Theorem 5.1). Second, we relate the inferred representation/reconstruction of test examples to the training data. We highlight several interpretable features of the unrolled dictionary learning network. Specifically, we perform analysis that provide insights into questions such as why am I learning a particular feature in the dictionary? or from what part of the training set or an image I am learning that feature? (Figure 8). Moreover, we provide an explanation of the relation between the new test image denoised/reconstructed through the network and the training dataset. The model provides insights on how training images are used to reconstruct a new test image (Figure 9) or how the test image picks up training images that have a similar representation to itself to reconstruct (Figure 10).
2 Related Works

There is vast literature on the theoretical convergence of dictionary learning. Spielman et al. (2012) proposed a factorization method to recover the dictionary in the undercomplete setting (i.e., $p \leq m$). Barak et al. (2015) proposed to solve dictionary learning via sum-of-squares semidefinite program. K-SVD (Aharon et al., 2006) and MOD (Engan et al., 1999) are popular greedy approaches. Alternating-minimization-based methods have been used extensively in theory and practice (Jain et al., 2013; Agarwal et al., 2014; Arora et al., 2014). Recent work has incorporated gradient-based updates into alternating minimization (Chatterji & Bartlett, 2017; Arora et al., 2015; Ramhhatla et al., 2018). Chatterji & Bartlett (2017) provided a finite sample analysis and convergence guarantees when updating the dictionary using the analytic gradient. Arora et al. (2015) proposed neurally plausible sparse coding approaches with analytic gradients. Another work focused on online dictionary learning (Mairal et al., 2009a) with an unbiased gradient updates (Rambhatla et al., 2018). Arora et al. (2015) discussed methods to reduce the bias of dictionary estimate, and Rambhatla et al. (2018) showed how to reduce bias in code and dictionary estimates. A common feature in the above-mentioned work is the use of analytic gradients, i.e., explicitly designing gradient updates independent of the sparse coding step and not utilizing automatic gradients with deep learning optimizers. A theoretical analysis of backpropagation for dictionary learning exists only for shallow autoencoders (Rangamani et al., 2018; Nguyen et al., 2019).

The theoretical analysis of unrolled neural networks has mainly analyzed the convergence speed of variants of LISTA (Gregor & LeCun, 2010), where the focus is on sparse coding (i.e., the encoder) not dictionary learning (Sprechmann et al., 2012; Xin et al., 2016; Moreau & Bruna, 2017; Giryes et al., 2018; Chen et al., 2018; Liu & Chen, 2019; Ablin et al., 2019). Moreau & Bruna (2017) showed that upon successful factorization onto the feasible set by performing $D\rightarrow \infty$ errors locally in each step of an alternating minimization and did not provide errors concerning gradients for model recovery, Ablin et al. (2020) and Malézieux et al. (2022) studied the asymptotic gradient approximation in the early regime of unrolling for dictionary learning. Unlike our work, where we evaluate the smooth and differentiable functions. Moreover, Malézieux et al. (2022) studied the stability of gradient backpropagation but still assumed the knowledge of the dictionary at the decoder. Ablin et al. (2020) compared analytic and automatic gradient estimators of min-min optimizations with smooth and differentiable functions. Moreover, Malézieux et al. (2022) studied the stability of gradient approximation in the early regime of unrolling for dictionary learning. Unlike our work, where we evaluate the gradients for model recovery, Ablin et al. (2020) and Malézieux et al. (2022) studied the asymptotic gradient errors locally in each step of an alternating minimization and did not provide errors concerning $z^*$ or $D^*$.

3 Preliminaries

Given $n$ independent samples, dictionary learning aims to minimize the empirical risk, i.e.,

$$\min_{D\in D} \mathcal{R}_n(D) \quad \text{with} \quad \mathcal{R}_n(D) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell_{x^i}(D)$$

where $\lim_{n\to\infty} \mathcal{R}_n(D) = \mathbb{E}_{x\in \mathcal{X}} [\ell_x(D)]$ a.s. To prevent scaling ambiguity between the code $z$ and dictionary $D$, it is common to constrain the norm of the dictionary columns. Hence, we define the set of feasible solutions for the dictionary as $D \triangleq \{D \in \mathbb{R}^{m \times p} \text{ s.t. } \forall j \in \{1, 2, \ldots, p\}, \|D_j\|_2^2 \leq 1\}$. We can project estimates of $D$ onto the feasible set by performing $D_{jj} \leftarrow 1/\max(\|D_{jj}\|_2, 1) D_{jj}$, either at every update or at the end of training. We assume certain properties on the data, specifically its domain (Assumption 3.1), energy (Assumption 3.2), code distribution (Assumption 3.3), and generating dictionary (Assumption 3.4).

**Assumption 3.1** (Domain signals). $\mathcal{X}$ and $\mathcal{D}$ are both compact convex sets.

**Assumption 3.2** (Bounded signals). $\exists M > 0 \text{ s.t. } \|x\|_2 < M \forall x \in \mathcal{X}$.

**Assumption 3.3** (Code distribution). The code $z^*$ is at most $s$-sparse with the support $S^* = \text{supp}(z^*)$. Each element in $S^*$ is chosen from the set $[1, p]$, uniformly at random without replacement. $p_i = P(i \in S^*) = \Theta(s/p)$, and $p_{ij} = P(i, j \in S^*) = \Theta(s^2/p^2)$. Given the support, $z^*_s$ is i.i.d, has symmetric probability distribution density function, $\mathbb{E}[z^*_s | i \in S^*] = 0$ and $\mathbb{E}[z^*_s z^*_s^T | S^*] = I$. Moreover, the non-zero entries of the code are sub-Gaussian and lower bounded, i.e., $|z^*_{(i,j)}| \geq C_{\text{min}}$ where $0 < C_{\text{min}} \leq 1$. 


The dictionary closeness at every update is denoted by \(\parallel D_i^j - D_i^* \parallel_2 \leq \delta_j \forall j\). Furthermore, \(\delta_j = O^*(1/\log p)\).

To achieve model recovery using gradient descent, we assume an appropriate dictionary initialization, i.e.,

**Assumption 3.5** (Dictionary closeness). The initial dictionary \(D^{(0)}\) is \((\delta_0, 2)\)-close to \(D^*\) (see Definition A.2). The dictionary closeness at every update is denoted by \(\parallel D_j^i - D_j^* \parallel_2 \leq \delta_1\). Furthermore, \(\delta_1 = O^*(1/\log p)\).

Arora et al. (2015) proposed a dictionary initialization method offering \((\delta, 2)\)-close to \(D^*\) for \(\delta = O^*(1/\log m)\). The method is based on pairwise reweighting of samples \(\{x^i\}_{i=1}^p\) from the generative model (2), and does not require access to \(D^*\). In addition, Rambhatla et al. (2018) utilize dictionary closeness assumptions and such dictionary initialization for their theoretical analysis. Moreover, Agarwal et al. (2017) proposed a clustering approach to find a close initial estimate of the dictionary.

Given the \(\mu\)-incoherence of \(D^*\) (Assumption 3.4) and \(\delta\)-closeness of the dictionary, \(D^{(i)}\) is \(\mu_1\)-incoherent, i.e.,

**Lemma 3.1** (\(\mu_1\)-incoherence). \(D^{(i)}\) is \(\mu_1\)-incoherent where \(\mu_1 = \mu + 2\sqrt{m}\delta_1\).

The recurrent encoder and decoder, which perform the computations shown in Algorithm 2, use the loss \(L \parallel x \parallel_1\) and proximal operator \(P_\delta(v) \triangleq \text{sign}(v) \max(|v| - b, 0)\) for the \(\ell_1\) norm \(h: \mathbb{R}^p \rightarrow \mathbb{R}\). The encoder implements ISTA (Daubechies et al., 2004; Blumensath & Davies, 2008) with step size \(\alpha\), assumed to be less than \(1/s_{\max}(D)\). With infinite encoder unrolling, the encoder’s output is the solution to the lasso (1), following the optimality condition (Lemma A.3) where we denote \(f_\alpha(z, D) \triangleq L_\alpha(z, D) + h(z)\). One immediate observation is that \(\lambda \geq \|D^T x\|_\infty \Leftrightarrow \{0\} \in \arg\min f_\alpha(z, D)\). We assume \(\lambda < \|D^T x\|_\infty\). We specify in Theorem 4.1 and Theorem 4.2 the conditions on \(\lambda\) for every encoder iteration to ensure support recovery and its preservation through the encoder. In case of a constant \(\lambda\) across encoder iterations while using \(D^*\) as the dictionary (i.e., sparse coding using \(\ell_1\) norm), the network recovers a biased code \(\hat{z}^*\). We denote this amplitude error in the code by \(\hat{\delta}^* \triangleq \|\hat{z}^* - z^*\|_2\) which is small and goes to zero with \(\lambda\) decaying through the encoder.

In \(\hat{\delta}^* \triangleq \|\hat{z}^* - z^*\|_2\) we assume the solution to (1) is unique; sufficient conditions for uniqueness in the overcomplete case (i.e., \(p > m\)) are extensively studied in the literature (Wainwright, 2009; Candès & Plan, 2009; Tibshirani, 2013). Tibshirani (2013) discussed that the solution is unique with probability one if entries of \(D\) are drawn from a continuous probability distribution (Tibshirani, 2013) (Assumption 3.6). This assumption implies that \(D^T D\) is full-rank. We argue that as long as the data \(x \in X\) are sampled from a continuous distribution, this assumption holds for the entire learning process. The preservation of this property is guaranteed at all iterations of the alternating minimization proposed in (Agarwal et al., 2014). Moreover, this assumption has been previously considered in analyses of unrolled sparse coding networks (Ablin et al., 2019; Malézieux et al., 2022) and can be extended to \(\ell_1\)-based optimization problems (Tibshirani, 2013; Rosset et al., 2004).

**Assumption 3.6** (Lasso uniqueness). The entries of the dictionary \(D\) are continuously distributed. Hence, the minimizer of (1) is unique, i.e., \(\hat{z} = \arg\min f_\alpha(z, D)\) with probability one.

**Lemma 3.2** states the fixed-point property of the encoder recursion (Parikh & Boyd, 2014). Given the definitions for Lipschitz and Lipschitz differentiable functions, (Definitions A.3 and A.4), the loss \(L\) and function \(h\) satisfy following Lipschitz properties.

**Lemma 3.2** (Fixed-point property of lasso). Given \(L_\alpha(z, D) \triangleq \frac{1}{2}\|x - Dz\|_2^2, D\), and Assumption 3.2, the loss is Lipschitz differentiable. Let \(L_{11} \) and \(L_{21}\) denote the Lipschitz constants of the first derivatives \(\nabla_{\alpha} L_\alpha(z, D)\) and \(\nabla_2 L_\alpha(z, D)\). \(L_{11}\) and \(L_{21}\) denote the Lipschitz constants of the second derivatives \(\nabla^2_{\alpha} L_\alpha(z, D)\) and \(\nabla^2_{21} L_\alpha(z, D)\) w.r.t \(z\). Let \(\nabla_{\alpha} L_\alpha(z, D)\) be \(L_{11}\)-Lipschitz w.r.t \(D\), and we denote the Lipschitz constant of \(\nabla_{\alpha} L_\alpha(z, D)\) \(L_{11}\) w.r.t \(D\) by \(L_{11D}\) and \(L_{21D}\) respectively.

**Lemma 3.3** (Lipschitz differentiable least squares). Given \(L_\alpha(z, D) \triangleq \frac{1}{2}\|x - Dz\|_2^2, D\), and Assumption 3.2, the loss is Lipschitz differentiable. Let \(L_{11} \) and \(L_{21}\) denote the Lipschitz constants of the first derivatives \(\nabla_{\alpha} L_\alpha(z, D)\) and \(\nabla_2 L_\alpha(z, D)\). \(L_{11}\) and \(L_{21}\) denote the Lipschitz constants of the second derivatives \(\nabla^2_{\alpha} L_\alpha(z, D)\) and \(\nabla^2_{21} L_\alpha(z, D)\) w.r.t \(z\). Let \(\nabla_{\alpha} L_\alpha(z, D)\) be \(L_{11}\)-Lipschitz w.r.t \(D\), and we denote the Lipschitz constant of \(\nabla_{\alpha} L_\alpha(z, D)\) \(L_{11}\) w.r.t \(D\) by \(L_{11D}\) and \(L_{21D}\) respectively.

**Lemma 3.4** (Lipschitz proximal). Given \(h(z) = \lambda \|z\|_1\), its proximal operator has bounded sub-derivative, i.e., \(\|\partial P_\lambda(z)\|_2 \leq c_{\text{prox}}\).
Algorithm 1: Classical alternating-minimization-based dictionary learning using lasso (1).

**Initialize:** Samples \( \{x^i\}_{i=1}^n \in X \), initial dictionary \( D^{(0)} \).

**Repeat:** \( l = 0, 1, \ldots \), number of epochs

**Sparse coding step:** \( z^{(i)} = \arg \min_z L_{\psi}(z, D^{(i)}) + h(z), \quad \text{for } i \in [1, n] \)

**Dictionary update:** \( D^{(l+1)} = D^{(l)} - \eta \check{g}^{(l)} \) where \( \check{g}^{(l)} \triangleq \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(z^{(i)}, D^{(l)}) \)


**Initialize:** Samples \( \{x^i\}_{i=1}^n \in X \), initial dictionary \( D^{(0)} \), and \( z_0 = 0 \).

**Repeat:** \( l = 0, 1, \ldots \), number of epochs

**Forward pass:** (for \( i \in [1, n] \))

Encoder: \( z_t^{(l)} = \Phi(x_t^{(l)}, D^{(l)}) = P_{x\lambda}(z_t^{(l)} - \alpha \nabla_1 L_{\psi}(z_t^{(l)}, D^{(l)})) \) (repeat for \( T \))

Decoder: \( \hat{x}^{(l)} = D^{(l)} z_T^{(l)} \) (6)

**Backward pass:** \( D^{(l+1)} = D^{(l)} - \eta g_T^{(l)} \) where \( g_T^{(l)} \) is either of

- \( g_T^{(l)} \text{-dec} \triangleq \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(z_T^{(l)}, D^{(l)}) \)
- \( g_T^{(l)} \text{-ae-lasso} \triangleq \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(z_T^{(l)}, D^{(l)}) + J_T^{(l)} + \nabla_1 L_{\psi}(z_T^{(l)}, D^{(l)}) + h(z_T^{(l)}) \) (7)
- \( g_T^{(l)} \text{-ae-ils} \triangleq \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(z_T^{(l)}, D^{(l)}) + J_T^{(l)} + \nabla_1 L_{\psi}(z_T^{(l)}, D^{(l)}) \)

See Definition 4.1 for \( J_T^+ \).

### 4 Unrolled Dictionary Learning

The gradients defined in PUDLE (Algorithm 2) can be compared against the local direction at each update of classical alternating-minimization (Algorithm 1). Assuming there are infinite samples, i.e.,

Best local direction: \( \check{g} \triangleq \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(\hat{z}^i, D) = E_{x \in X} [\nabla_2 L_{\psi}(\hat{z}, D)] \) (4)

where \( \hat{z} = \arg \min_{z \in \mathbb{R}^p} L_{\psi}(z, D) + h(z) \). Additionally, to assess the estimators for model recovery, hence dictionary learning, we compare them against gradient pointing towards \( D^* \), namely

Desired global gradient for \( D^* \): \( g^* \triangleq \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \nabla_2 L_{\psi}(z^*, D) = E_{x \in X} [\nabla_2 L_{\psi}(z^*, D)] \). (5)

To see why the above is the desired direction, \( (z^*, D^*) \) is a critical point of the loss \( L \) which reaches zero for data following the model (2). Hence, to reach \( D^* \in \arg \min_{D \in \mathcal{X}} E_{x \in X} [L_{\psi}(z^*, D)] \), we move towards the direction minimizing the loss in expectation. Specifically, using the gradient \( \nabla_2 L_{\psi}(z^*, D) = -(x - Dz^*)z^*T = (D - D^*)z^*z^*T \) as a descent direction, we move from \( D \) toward \( D^* \) modulo the code presence matrix \( z^*z^*T \). Given these directions, we analyze the error of the gradients \( g_t^{\text{dec}} \), \( g_t^{\text{ae-lasso}} \), and \( g_t^{\text{ae-ils}} \) assuming infinite samples. In local analysis, we compare the code and gradient estimates to the lasso optimization in each update of the alternating minimization. In global analysis, we evaluate the performance in recovery of the ground-truth code \( z^* \) and the dictionary \( D^* \). In this regard, we first study the forward pass.

### 4.1 Forward pass

We show convergence results in the forward pass for \( z \) and the Jacobian, i.e.,

**Definition 4.1 (Code Jacobian).** Given \( D \), the Jacobian of \( z_t \) is defined as \( J_t \triangleq \frac{\partial z_t}{\partial D} \) with adjoint \( J_t^+ \).

The forward pass analyses give upper bounds on the error between \( z_t \) and \( \hat{z} \) and the error between \( J_t \) and \( \hat{J} \) as a function of unrolled iterations \( t \). We define \( J \) as following: considering the function \( z \to L_{\psi}(z, D) + h(z) \), \( \hat{z}(D) \) is its minimizer and \( \hat{J} = \frac{\partial \hat{z}(D)}{\partial D} \). We will require these errors in Section 4.2, where we analyze the gradient estimation errors. Similar to (Chatterji & Bartlett, 2017), the error associated with \( g_t^{\text{dec}} \) depends on
the code convergence. Unlike $g^{\text{dec}}$, the convergence of backpropagation with gradient estimates $g^{\text{ae-lasso}}$ and $g^{\text{ae-bs}}$ relies on the convergence properties of the code and the Jacobian (Ablin et al., 2020). Forward-pass theories are based on studies by Gilbert (1992) on the convergence of variables and their derivatives in an iterative process governed by a smooth operator (Gilbert, 1992). Moreover, Hale et al. (2007) studied the convergence analysis of fixed point iterations for $\ell_1$ regularized optimization problems (Hale et al., 2007).

Support recovery and preservation We re-state a result from (Hale et al., 2007) on support selection.

**Proposition 4.1** (Finite-iteration support selection). Given Assumption 3.6, let $\hat{z} = \arg \min f_\alpha (z, D)$ with $S \triangleq \text{supp}(\hat{z})$. There exists a $B > 0$ such that $\text{supp}(z_t) = S, \forall t > B$.

This means the unrolled encoder identifies the support in finite iterations. Support recovery in finite iterations has been studied in the literature for LISTA (Chen et al., 2018), Step-LISTA (Ablin et al., 2019), and shallow autoencoders (Arora et al., 2015; Rangamani et al., 2018; Nguyen et al., 2019; Tolosshams et al., 2020). We show that under proper initialization of the dictionary, the encoder achieves linear convergence. Arora et al. (2015) discussed some appropriate initialization which is used by Rambhatla et al. (2018). Given initial closeness $\delta_0$, the encoder selects and recovers the correct signed support of the code with high probability in one iteration $B = 1$ (Theorem 4.1), and the iterations preserve the correct support (Theorem 4.2). In spite of slow convergence of ISTA Liang et al. (2014), support recovery after one iteration in unrolled networks is studied in the literature (Arora et al., 2015; Rambhatla et al., 2018; Chen et al., 2018; Nguyen et al., 2019).

**Theorem 4.1** (Forward pass support recovery). Given Assumptions 3.3 and 3.4, suppose $D^{(l)}$ is $\delta_t = O^*(1/\sqrt{\log p})$ close to $D^*$. If $s = O^*(\sqrt{m}/\mu \log m)$, and $\mu = O(\log m)$, then with probability of at least $1 - \epsilon^{(l)}_{\text{supp-rec}}$, the choice of $\lambda_0 = C_{\min}/4$ recovers the support of the code $z^*$ in one encoder iteration, i.e., $\text{sign}(S_{\alpha, \lambda_0}(\alpha D^{(l)} T x)) = \text{sign}(z^*)$, where $\epsilon^{(l)}_{\text{supp-rec}} = 2p \exp\left(-\frac{C_{\min}^2}{\sigma^2} \delta_t^2\right)$.

**Theorem 4.2** (Forward pass support preservation). Given Assumptions 3.3 and 3.4, suppose $D^{(l)}$ is $\delta_t = O^*(1/\log p)$ close to $D^*$. If $s = O^*(\sqrt{m}/\mu \log m)$, $\mu = O(\log m)$, and the regularizer and step size are chosen such that $\lambda_t^{(l)} = \frac{\mu}{\sqrt{m}} \|z^* - z_t\|_1 + a_{\gamma} = O\left(\frac{\log m}{\sqrt{m}}\right)$ and $\alpha^{(l)} \leq 1 - \frac{2\lambda_t^{(l)} - (1 - \frac{\epsilon_t}{\sqrt{C_{\min}}})^2}{\lambda_t^{(l)} + 1}$, then with probability of at least $1 - \epsilon^{(l)}_{\text{supp-pres}}$, the support, recovered at the first iteration, is preserved through the encoder iterations. We have $a_{\gamma} = O(\sqrt{s \delta_t})$ and $\epsilon^{(l)}_{\text{supp-pres}} := \epsilon^{(l)}_{\text{supp-rec}} + \epsilon^{(l)}_{\gamma} = 2p \exp\left(-\frac{C_{\min}^2}{\sigma^2} \delta_t^2\right) + 2s \exp\left(-\frac{1}{C_{\min}^2 s^2}\right)$.

The support preservation conditions on $\lambda_t$ and $\alpha$ introduce two insights. First, with an increase of $t$, the code error decrease, hence the lower bound on $\lambda_t$. Second, the decay of $\lambda_t$ as the encoder unrolls increases the upper bound on $\alpha$. Hence, we suggest a decaying strategy in values of $\lambda_t$ as $t$ increases.

The utilization of knowledge of the code error, as we do in Theorem 4.2, to set the proper thresholding/bias/regularization parameters ($\lambda$) constitutes a fairly standard practice. Below we discuss similar results in the literature. For the preservation of correct signed-support in a sparse coding network, Rambhatla et al. (2018) provided a proper thresholding value at every iteration as a function of the $\ell_1$-norm of the code error with respect to a ground-truth code; they additionally demonstrated an upper bound on the estimate of the code coefficients as a function of dictionary closeness. Moreover, Nguyen et al. (2019) used information on the range of ground-truth code to choose proper biases in their neural network to guarantee support recovery. Chen et al. (2018) similarly provided an upper bound on the bias of their unrolled sparse coding network at every layer as a function of $\ell_2$-norm error between the code estimate at the layer and the ground-truth code. Overall, the error between a code estimate and the ground-truth code appearing in the lower bound on $\lambda_t^{(l)}$ can further simplified into terms related to terms such as the dictionary closeness $\delta_t$, code sparsity. For example, Chatterji & Bartlett (2017), for their particular sparse coding algorithm, provided $\ell_\infty$-norm upper bound as a function of terms such as code sparsity, data dimensionality, code range, and dictionary error.

**Code convergence and error** Given the support recovery and its preservation, the encoder convergence studied in (Malézieux et al., 2022) can achieve linear convergences after its first iteration. We re-state this result on the rate of convergence of the encoder in Theorem 4.3. We drop the superscript $(l)$ to simplify the notation.
Theorem 4.3 (Local forward pass code convergence). Given the encoder 
\( z_{t+1} = \Phi(z_t, D) \), Assumption 3.6, Lemmas 3.2, A.1 and A.2, 
then \( \exists \rho < 1, B > 0 \) s.t. \( \| z_t - \hat{z} \|_2 \leq O(\rho^t) \) \( \forall t > B \), where \( \hat{z} \) is the 
unique minimizer of lasso (1). Furthermore, given Theorem 4.1 and 
Theorem 4.2, \( B = 1 \).

Theorem 4.3 shows that in PUDLE, \( z_t \) converges to \( \hat{z} \) at a linear 
rate eventually after a certain number of unrolling (Figure 2). The local 
linear convergence of ISTA and FISTA (Beck & Teboulle, 2009) 
(with global rates of \( O(1/t) \) and \( O(1/t^2) \)) in the neighbourhood of a 
fixed-point is studied in (Tao et al., 2016). The speed of convergence 
depends on when support selection happens (Proposition 4.1) (Bredies 
& Lorenz, 2008; Zhang et al., 2017b; Liang et al., 2014). We showed 
that under mild assumptions, the support is selected and recovered 
after one encoder iteration. In addition to local convergence, we focus on 
recovery of \( z^* \) and show error on the unrolled code coefficients \( z_{t,j}^{(l)} \) 
with respect to ground-truth \( z_{j}^{(l)} \) as \( t \) increases. In Theorem 4.4, we consider the case where \( \lambda_t \) at layer \( t \) is set 
to according to Theorem 4.2; the bias decreases as the code error decreases 
among the layers and dictionary updates. We provide an upper bound on 
the coefficients errors as a function of code sparsity, dictionary error, and 
a unrolling error \( e_{t,j}^{(l)unroll} \). The unrolling error goes to zero for appropriately large \( t \). Moreover, Theorem 4.5 studies the case 
where the bias is fixed across the layers. In this scenario, we observe an 
additional term of \( \lambda^{fixed} \) in the upper bounds on the code coefficients error; this term shows that the code error when we strictly perform \( \ell_1 \)-norm 
based sparse coding does not go to zero. We refer to this error as an amplitude bias estimate error.

Theorem 4.4 (Global forward pass code error with variable \( \lambda_t \)). Given Assumptions 3.3 and 3.4, suppose \( D^{(l)} \) 
is \( \mu \)-incoherent and \( \delta_t = O^*(1/\log p) \) close to \( D^* \). If \( s = O^*(\sqrt{m}/\mu \log m) \), \( \mu = O(\log m) \), and the 
regularizer and step size are chosen such that \( \lambda_t^{(l)} = \frac{m}{\sqrt{m}} ||z^*-z||_1 + a_\gamma = \Omega(\frac{m \log m}{\sqrt{m}}) \) and \( \alpha^{(l)} \leq 1 - \frac{2\lambda_t^{(l)} - (1 - \frac{\sqrt{\log m}}{\sqrt{m}})}{\lambda_t^{(l)-1}} \), 
then with probability of at least \( 1 - \epsilon_{supp-pres,j}^{(l)} \), for \( j \in supp(z^*) \), the code coefficient error is 
\[ ||z_{t,j}^{(l)} - z_j^{*}||_2 \leq O(\sqrt{s}||D_j^{(l)} - D_j^*||_2) + e_{t,j}^{(l)unroll} \] 
(8)
and 
\[ z_{T,j}^{(l)} = z_j^{*}(1 - \beta_j^{(l)}) + c_{T,j}^{(l)} \] 
(9)
where \( e_{t,j}^{(l)unroll} := 2(s-1)\alpha m \sqrt{m} \max_{i} |z_{t,j}^{(l)} - z_{j}^{*}| \delta_{a,1-t} + |z_{0,j} - z_{j}^{*}| \delta_{a,t} \), \( \delta_{a,1-t} := (1 - \alpha + 2\alpha \frac{m}{\sqrt{m}}) \), \( \delta_{a,t} := (1 - \alpha + 2\alpha \frac{m}{\sqrt{m}}) \).

Theorem 4.5 (Global forward pass code error with fixed \( \lambda_t \)). Given Assumptions 3.3 and 3.4, suppose \( D^{(l)} \) is \( \mu \)-incoherent and \( \delta_t = O^*(1/\log p) \) close to \( D^* \). If \( s = O^*(\sqrt{m}/\mu \log m) \), \( \mu = O(\log m) \), and the 
regularizer and step size are chosen such that \( \lambda_t^{(l)} = \lambda^{fixed} = \frac{m}{\sqrt{m}} ||z^*-z_0||_1 + a_\gamma = \Omega(\frac{m \log m}{\sqrt{m}}) \) and 
\( \alpha^{(l)} \leq 1 - \frac{2\lambda_t^{(l)} - (1 - \frac{\sqrt{\log m}}{\sqrt{m}})}{\lambda_t^{(l)-1}} \), then with probability of at least \( 1 - \epsilon_{supp-pres,j}^{(l)} \), for \( j \in supp(z^*) \), the code coefficient error is 
\[ ||z_{t,j}^{(l)} - z_j^{*}||_2 \leq O(\sqrt{s}||D_j^{(l)} - D_j^*||_2) + e_{t,j}^{(l)unroll,fixed} + \lambda^{fixed} \] 
(10)
and 
\[ z_{T,j}^{(l)} = z_j^{*}(1 - \beta_j^{(l)}) + c_{T,j}^{(l)} \] 
(11)
where \( e_{t,j}^{(l)unroll,fixed} := (s-1)\alpha m \sqrt{m} \max_{i} |z_{t,j}^{(l)} - z_{j}^{*}| \delta_{a,1-t}^{fixed} + |z_{0,j} - z_{j}^{*}| \delta_{a,t}^{fixed} \), \( \delta_{a,1-t}^{fixed} := (1 - \alpha + 2\alpha \frac{m}{\sqrt{m}}) \), \( \delta_{a,t}^{fixed} := (1 - \alpha + 2\alpha \frac{m}{\sqrt{m}}) \).
Aside from code estimation where the network parameters (e.g., regularization and step size) are finely tuned according to support recovery and preservation conditions (Theorems 4.1 and 4.2), we provide a general upper bound on the error between the converged code and $z^*$; the bound can be decomposed into two terms of the dictionary error and the biased amplitude estimate of the code.

**Theorem 4.6** (Global forward pass code error). Let $\hat{z}$ be the fixed-point of the encoder with iterations $z_{t+1} = \Phi(z_t, D)$. Given Assumption 3.6, Lemmas 3.2, A.1 and A.2, we have $\|\hat{z} - z^*\|_2 \leq O(||D - D^*||_2 + \delta^*)$, where $\delta^* = ||\hat{z}^* - z^*||_2$, $\hat{z}$ is the unique minimizer of lasso (1) given the dictionary $D$, $z^*$ is the unique minimizer of lasso (1) given the dictionary $D^*$, and $z^*$ is the ground-truth code.

This general decomposition is to emphasize that aside from the current estimate of the dictionary, the code error is a function of the forward pass algorithm used to solve the sparse coding problem. Specifically, the upper bound states that at the best scenario where there is access to the generating dictionary $D^*$, the forward pass solving lasso with fixed $\lambda$ still gives a biased amplitude estimate of $z^*$. Overall, the assumptions to get this bound are mild; the bound is valid independent of successful support recovery or data distribution. With incorporation of data distribution and conditions stated in Theorem 4.1 and Theorem 4.2, the upper bound $\delta^*$ can be replaced with terms involving $\lambda$, and reaches at zero as $\lambda$ decays across forward iterations.

**Jacobian convergence and error** Following properties similar to those used in Theorem 4.3, and assuming $J_t$ is bounded (Assumption 4.1), we show in Theorem 4.7 that, as the PUDLE unrolls, the code Jacobian $J_t$ converges to $\hat{J}$, the Jacobian of the solution of the lasso. The convergence of the Jacobian of proximal gradient descent is also studied in (Bertrand et al., 2021) for hyperparameter selection through implicit differentiation (Bengio, 2000), where the Jacobian is taken w.r.t to the hyperparameter $\lambda$ as opposed to $D$.

**Assumption 4.1** (Bounded Jacobian). The Jacobian is bounded, i.e., $\exists M_j > 0$, s.t. $\|J_t\|_2 \leq M_j \forall t$.

**Theorem 4.7** (Local forward pass Jacobian convergence). Given the recursion $z_{t+1} = \Phi(z_t, D)$, and $\tilde{z}$ the unique minimizer of lasso with Jacobian $\tilde{J}$, then $\exists \rho < 1, B > 0$ s.t. $\|J_t - \tilde{J}\|_2 \leq O(\rho t^B) \forall t > B$. Furthermore, given Theorem 4.1 and Theorem 4.2, $B = 1$.

The forward pass code and Jacobian convergences after support selection is similar to the results from (Mâlézieux et al., 2022). The highlights of our finding are that the order of upper bound convergences can be achieved from the first iteration of the encoder. In other words, we specify, in Theorem 4.1 and Theorem 4.2, the dictionary and data conditions such that the support can be recovered with $B = 1$. This resolves the instability issue discussed by Malézieux et al. (2022) in computation of the gradient $g_{ae}^{lasso}$ outside of the support. Finally, we show that the global Jacobian error is in the order of dictionary error.

**Theorem 4.8** (Global forward pass Jacobian error). Let $\hat{z}$ be the fixed-point of the encoder with iterations $z_{t+1} = \Phi(z_t, D)$. Given Assumption 3.6, Lemmas 3.2, A.1 and A.2, we have $\|\hat{J} - J^*\|_2 \leq O(||D - D^*||_2 + \delta_j^*)$, where $\delta_j^* := \|J^* - \hat{J^*}\|_2$, and $\hat{J}$, $J^*$ and $J^*$ are Jacobians corresponding to $\hat{z}$, $z^*$ and $z^*$.

### 4.2 Backward pass

We show two results for local gradient $\hat{g}$ and global gradient $g^*$ convergence. The goal is not to provide a finite sample analysis but to emphasize the relative differences between the gradients in Algorithm 2. The impact of gradient error for parameter estimation in the convex setting has been studied by Devolder et al. (2013) indicating that the convergence to the parameter’s neighbourhood is dictated by the gradient error (Devolder et al., 2013; 2014). As dictionary learning is a bi-convex problem, findings of Devolder et al. (2013) hold as well for better estimation of the local dictionary at every step of alternating minimization. Moreover, Arora et al. (2015), provided a detailed analysis of sparse coding and various gradient estimations for dictionary learning, showing that by computing a more accurate gradient at every step of the alternating minimization scheme, the dictionary estimates
converge to a closer neighbourhood of $D^\ast$. Overall, the intuition is that the size of the gradient error dictates the size of the neighbourhood of the dictionary within which one can guarantee convergence. We argue that the method with lower gradient error recovers the dictionary better.

**Local gradient estimations**  We highlight the effect of finite unrolling on the gradient for parameter estimation (Ablin et al., 2020). Theorem 4.9 shows the convergence rate of gradients to $\hat{g}$, determining the similarity of PUDLE and Algorithm 1.

**Theorem 4.9** (Local convergence of gradients). Given the forward pass convergence results (Theorems 4.3 and 4.7), $\exists \rho < 1, B > 0$ such that $\forall t > B$, the errors of gradients defined in Algorithm 2 w.r.t $\hat{g}$ (4) satisfy

\[
\|g_{t}^{\text{dec}} - \hat{g}\|_2 \leq O(\rho^t) \\
\|g_{t}^{\text{ae-lasso}} - \hat{g}\|_2 \leq O(t\rho^2 t) \\
\|g_{t}^{\text{ae-ls}} - \hat{g}\|_2 \leq O(t\rho^3 + M_j\sqrt{s}).
\]

Moreover, the order of upper bounds is tight (see Lemma A.4).

First, upper bounds on the errors related to $g_{t}^{\text{dec}}$ and $g_{t}^{\text{ae-lasso}}$ go to zero as $t$ increases. Hence, both gradients converge to $\hat{g}$. This means that asymptotically as $t$ increases, training PUDLE with $g_{t}^{\text{dec}}$ and $g_{t}^{\text{ae-lasso}}$ is equivalent to classical alternating-minimization (Algorithm 1). Second, as $t$ increases, $g_{t}^{\text{ae-lasso}}$ has faster convergence than $g_{t}^{\text{dec}}$. Lastly, $g_{t}^{\text{ae-ls}}$ is a biased estimator of $\hat{g}$ (Figure 3). The convergence results on the error $\|g_{t}^{\text{ae-lasso}} - \hat{g}\|_2$ is previously studied by Malézieux et al. (2022).

Given the above convergence results, one may conclude that $g_{t}^{\text{ae-lasso}}$ should be used for dictionary recovery. However, we show next that for dictionary recovery, the gradient $g_{t}^{\text{ae-lasso}}$, used by Malézieux et al. (2022), is indeed a biased estimator of the global gradient $g^\ast$ for recovery of $D^\ast$. We decrease this bias by replacing $g_{t}^{\text{ae-lasso}}$ with $g_{t}^{\text{ae-ls}}$ and show that $g_{t}^{\text{ae-ls}}$ results in a better recovery of $D^\ast$ than $g_{t}^{\text{ae-lasso}}$.

**Global gradient estimations**  Theorem 4.10 shows the global gradient errors w.r.t $g^\ast$ from (5). We omit the gradient $g_{t}^{\text{dec}}$, as it is asymptotically equivalent to $g_{t}^{\text{ae-lasso}}$. We study the errors in the limit to unrolling, i.e., as $t \to \infty$. This determines which PUDLE gradients recover $D^\ast$ better (Devolder et al., 2013; 2014).

**Theorem 4.10** (Global error of gradients). Given the convergence results from the forward pass, (Theorems 4.6 and 4.8), the errors of gradients defined in Algorithm 2 w.r.t global direction $g^\ast$ (defined in (5)) satisfy

\[
\|g_{\infty}^{\text{ae-lasso}} - g^\ast\|_2 \leq O(\|D - D^\ast\|_2^2 + \|D - D^\ast\|_2 + \hat{\delta}^\ast + \hat{\delta}^\gamma + M_j\lambda\sqrt{s}) \\
\|g_{\infty}^{\text{ae-ls}} - g^\ast\|_2 \leq O(\|D - D^\ast\|_2^2 + \|D - D^\ast\|_2 + \hat{\delta}^\ast + \hat{\delta}^\gamma).
\]

Several factors affect the order of upper bounds: the current estimate of the dictionary, code amplitude-bias error due to $\ell_1$ norm, and the usage of $\ell_1$ norm in the loss used for backpropagation. To study the bias in the gradient computation, let consider the scenario where $D = D^\ast$. We denote those gradients by superscript $D^\ast$. If the gradients are not biased, then the upper bounds should goes to zero. The gradient errors are

\[
\|g_{\infty}^{\text{ae-lasso},D^\ast} - g^\ast\|_2 \leq O(\hat{\delta}^\ast + \hat{\delta}^\gamma + M_j\lambda\sqrt{s}) \quad \text{and} \quad \|g_{\infty}^{\text{ae-ls},D^\ast} - g^\ast\|_2 \leq O(\hat{\delta}^\ast + \hat{\delta}^\gamma).
\]

For $g_{\infty}^{\text{ae-ls},D^\ast}$, the radius of the error ball is only a function of the amplitude error of the code estimated through lasso compare to the ground-truth code $x^\ast$. However, the error ball for the gradient $g_{\infty}^{\text{ae-lasso},D^\ast}$ includes an additional term concerning the usage of lasso loss containing the regularization term $\lambda$. This implies that the $D^\ast$ neighbourhood at which the gradient $g_{\infty}^{\text{ae-lasso},D^\ast}$ is guaranteed to converge to is smaller than of the $g_{\infty}^{\text{ae-lasso},D^\ast}$ (Figure 4a). Implications of such gradient estimation are seen in dictionary learning where $g_{\infty}^{\text{ae-ls}}$ recovers $D^\ast$ better (Figures 4b and 4c). In Figure 4b, the encoder unrolls for $T = 25$, hence the phenomenon of acceleration is seen in faster and better dictionary learning performance of $g_{\infty}^{\text{ae-lasso}}$ than $g_{\infty}^{\text{dec}}$. In Figure 4c where $T = 100$, similar performance of $g_{\infty}^{\text{dec}}$ and $g_{\infty}^{\text{ae-lasso}}$ illustrates their asymptotic equivalence as $t \to \infty$ (See Appendix for additional noisy dictionary learning experiments where the measurements $x$ are corrupted with zero-mean Gaussian noise such that the Signal-to-Noise-Ratio is approximately 12 SNR; in this setting, the aforementioned comparative analysis still holds.)
Figure 4: Results for PUDLE’s global convergence (Theorem 4.10) and dictionary learning.

Towards unbiased estimation  As long as $\lambda$ is fixed within PUDLE, all defined gradients remain biased estimators of $g^*$, due to the biased estimate of the code $z^*$ through $\ell_1$ norm. This bias exists while dictionary learning is performed strictly using lasso through Algorithm 1. Given the conditions on the regularizer in Theorem 4.2 which we discussed in Section 4.1 and the derived upper bounds in Theorem 4.10, we suggest the decaying of $\lambda$ across the encoder to reduce the gradient biases and improve dictionary learning. Next, we prove in Theorem 4.11 that PUDLE converges to $D^*$ if $\lambda$ decays across the layers $t$ according to Theorem 4.4. Moreover, Theorem 4.12 proves that if $\lambda$ stays fixed according to Theorem 4.5, then PUDLE only guarantees to converge to a close neighbourhood of the dictionary. In these analyses, we focus on $g_T^{\text{dec}}$. Furthermore, we show in Section 4.3 that by decaying $\lambda$ at each unrolled layer, the gradient bias vanishes, and we recover $D^*$.

Dictionary learning  Given the network parameters set by Theorem 4.4, Theorem 4.11 proves that using $g_T^{\text{dec}}$, PUDLE recovers the dictionary; the dictionary error contracts at every update. Moreover, Theorem 4.12 proves that as long as $\lambda$ stays fixed across the unrolled layers, PUDLE guarantees to converge to only $D^*$ neighbourhood characterized by the regularization parameter $\lambda$. These analyses requires for $D^{(t)}$ to maintain a closeness to $D^*$ which we provide a proof for in Lemma A.7. Hence, the dictionary closeness assumption (Assumption 3.5) stays valid.

Theorem 4.11 (Dictionary learning with variable $\lambda_t$). Given Assumptions 3.3 and 3.4, suppose $D^{(t)}$ is $\mu_1$-incoherent and $(\delta_t, 2)$-close to $D^*$ with $\delta_t = O^*(1/\log p)$. If $s = O(\sqrt{m})$, $\mu = O(\log m)$, learning rate is $\eta = O\left(\frac{1}{\sqrt{t}}\right)$, and the regularizer and step size are set according to Theorem 4.4, then for any dictionary update $l$ using $g_T^{\text{dec}}$, with probability of at least $1 - \epsilon_{\text{supp-pres}}^{(l)}$,\begin{equation}
\|D_j^{(l+1)} - D_j^*\|_2^2 \leq (1 - \psi)\|D_j^{(l)} - D_j^*\|_2^2 + \epsilon_{\text{supp-pres}}^{(l)}\end{equation}
where $\epsilon_{\text{supp-pres}}^{(l)} := \epsilon_{\text{supp-rec}}^{(l)} + \epsilon_{\gamma}^{(l)} = 2p \exp \left(\frac{-C^2_{\text{min}}}{\sqrt{\gamma}}\right) + 2s \exp \left(\frac{-1}{\sqrt{\delta_t}}\right)$.

Theorem 4.12 (Dictionary learning with fixed $\lambda_t$). Given Assumptions 3.3 and 3.4, suppose $D^{(t)}$ is $\mu_1$-incoherent and $(\delta_t, 2)$-close to $D^*$ with $\delta_t = O^*(1/\log p)$. If $s = O(\sqrt{m})$, $\mu = O(\log m)$, learning rate is $\eta = O\left(\frac{1}{\sqrt{t}}\right)$, and the regularizer $\lambda^{\text{fixed}}$ and step size are set according to Theorem 4.5, then for any dictionary update $l$ using $g_T^{\text{dec}}$, with probability of at least $1 - \epsilon_{\text{supp-pres}}^{(l)}$,\begin{equation}
\|D_j^{(l+1)} - D_j^*\|_2^2 \leq (1 - \psi)\|D_j^{(l)} - D_j^*\|_2^2 + \epsilon_{\lambda}^{(l)}\end{equation}
where $\epsilon_{\lambda}^{(l)} := \eta - \frac{2p}{s(1 - \beta_j^*)}\lambda^{\text{fixed}}2$, $\epsilon_{\text{supp-pres}}^{(l)} := \epsilon_{\text{supp-rec}}^{(l)} + \epsilon_{\gamma}^{(l)} = 2p \exp \left(\frac{-C^2_{\text{min}}}{\sqrt{\gamma}}\right) + 2s \exp \left(\frac{-1}{\sqrt{\delta_t}}\right)$.

4.3 Experiments

Dictionary learning  We focus on the performance of the best-performing gradient estimator $g_T^{\text{ae-ls}}$, and compare it with NOODL (Rambhatla et al., 2018), a state-of-the-art online dictionary learning algorithm, and SPORCO (Wohlberg, 2017), an alternating-minimization dictionary learning algorithm that uses lasso. NOODL, which uses iterative hard-thresholding (HT) for sparse coding and a gradient update
employing the code’s sign, has linear convergence upon proper initialization (Rambhatla et al., 2018). We note that the results from $g_t^{ae-lasso}$ are not shown, as the gradient computation was unstable (Malézieux et al., 2022). We emphasize that our proposed gradient $g_t^{ae-ls}$ does not suffer such instability. We train:

- $g_t^{ae-ls}$: $\lambda$ is fixed across iterations.
- $g_t^{ae-ls}$, decay: $\lambda$ decays (i.e., $\lambda_t = \lambda \nu^t$, with $0 < \nu < 1$) where $\nu$ decreases as training progresses.
- $g_t^{ae-ls}$, HT: $\mathcal{P}_{\lambda}(v)$ is replaced with $HT_b(v)$.

With HT, the sparse coding step reduces to that from NOODL. In this case, we highlight the difference between the gradient update of our method (backpropagation) with NOODL. We focus on convergence, as $\eta$ across methods is not comparable.

Figure 5 shows the convergence of $D \in \mathbb{R}^{1000 \times 1500}$ to $D^*$ when the code is 20-sparse (for other sparsity levels and details see Appendix C). A biased estimate of the code amplitudes results in convergence only to a neighbourhood of the dictionary (Rambhatla et al., 2018). This is observed in the convergence of $g_t^{ae-lasso}$ and SPORCO (final error is shown). The convergence of $g_t^{ae-ls}$ to a closer neighbourhood than SPORCO supports Theorem 4.10. Moreover, with decaying $\lambda$, the code bias vanishes, hence $g_t^{ae-lasso}$, decay and $g_t^{ae-lasso}$, HT converges to $D^*$ similar to NOODL.

![Figure 5: Dictionary convergences.](image)

Figure 5: Dictionary convergences.

(a) $g_t^{ae-lasso}$ training is not stable. (b) $g_t^{ae-ls}$ improvement is smoother than $g_t^{dec}$.

![Figure 6: Networks behaviour (test PSNR) during training as a function of epochs.](image)

Figure 6: Networks behaviour (test PSNR) during training as a function of epochs.

**Image denoising** To further highlight the advantage of $g_t^{ae-ls}$ over the other gradients, we compare them in a supervised task of image denoising. In addition to $g_t^{ae-ls}$, $g_t^{ae-lasso}$, and $g_t^{dec}$, we consider $g_t^{ae-lasso}$, HT where the proximal operator is replaced with HT. This is to compare with sparse coding scheme of NOODL. We do not compare against NOODL’s dictionary update, as this computation for two-dimensional convolutions is not straightforward. Prior works have shown that variants of PUDLE either rival or outperform state-of-the-art architectures (Simon & Elad, 2019; Tolooshams et al., 2020). Thus, we focus on a comparative analysis of the gradients. We trained on 432 and tested on 68 images from BSD (Martin et al., 2001). BSD dataset is a popular training dataset for denoising (Zhang et al., 2017a; Simon & Elad, 2019; Mohan et al., 2019). We used a convolutional dictionary and corrupted images with zero-mean Gaussian noise of standard deviation of 25 (see Appendix C for details). We initialized the dictionary filters by standard Normal distribution; this is to follow the norm in the deep learning literature and to demonstrate the practicality and usefulness of PUDLE in the absence of an initialization method. We evaluate the denoising performance of soft-thresholding using $\lambda$ and HT with $b$ in peak signal-to-noise-ratio (PSNR).

![Figure 7: PSNR improvement during training.](image)

Figure 7: PSNR improvement during training.
First, we highlight the stability of $g_t^{\text{ae-ls}}$ against $g_t^{\text{ae-lasso}}$. Figure 6a shows the network dynamics in terms of test PSNR as a function of epochs when $\lambda = 0.16$ for $g_t^{\text{ae-ls}}, g_t^{\text{ae-lasso}}, g_t^{\text{dec}}$ and $b = 0.05$ for $g_t^{\text{ae-ls}}, HT$. We observed that $g_t^{\text{ae-ls}}$ uses full backpropagation and stays stable. However, the training with $g_t^{\text{ae-lasso}}$ is not stable and unstable to perform denoising where the noisy PSNR is approximately 20 dB (Malézieux et al., 2022). Second, Figure 6b shows that compared to $g_t^{\text{dec}}$, the backpropagated gradients result in a smoother improvement during training. Moreover, Table 1 shows that the advantage of $g_t^{\text{ae-ls}}$ over $g_t^{\text{dec}}$ is not limited to dictionary learning and is seen in denoising. We have excluded the results for $g_t^{\text{ae-lasso}}$ from Table 1 as the network failed to denoise (see Figure 6a). Additionally, the superior performance of $g_t^{\text{ae-ls}}$ compared to $g_t^{\text{ae-ls}}, HT$ highlights the benefits of PUDLE (i.e., $\ell_1$-based unrolling) against HT used in NOODL.

Table 1: Denoising of BSD68. Reported numbers are mean (std) PSNR given three independent trials.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>PSNR [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$ 0.08</td>
</tr>
<tr>
<td>$g_t^{\text{dec}}$</td>
<td>24.21 (0.12)</td>
</tr>
<tr>
<td>$g_t^{\text{ae-ls}}$</td>
<td>24.79 (0.03)</td>
</tr>
<tr>
<td>$b$</td>
<td>0.02</td>
</tr>
<tr>
<td>$g_t^{\text{ae-ls}}, HT$</td>
<td>22.92 (0.07)</td>
</tr>
</tbody>
</table>

5 Interpretable Sparse Codes and Dictionary

One motivation behind using algorithm unrolling to design deep architectures is interpretability (Monga et al., 2019); they argue that the designed networks are interpretable as they capture domain knowledge via an optimization model. For example, Tolooshams et al. (2021a) takes advantage of the interpretability of learned weights in an unrolled dictionary learning network to solve spike sorting, an unsupervised source separation problem in computational neuroscience. Moreover, Kim et al. (2010) uses sparse coding to learn interpretable representations of human motions. However, none of the existing methods in the literature provide interpretability results that open the black-box network through building a mathematical relation between the learned dictionary, training data, and test representation/reconstruction. This section analyzes the interpretability of the unrolled sparse coding method in this context. We note that such mathematical relation and interpretability results also hold for dictionary learning. However, it is missing in the literature, irrespective of whether one uses an unrolling network. We provide the following theorem.

**Theorem 5.1** (Interpretable unrolled network). Consider the dictionary learning optimization of the form $\min Z, D \frac{1}{2} \|X - DZ\|_F^2 + \lambda \|Z\|_1 + \omega/2 \|D\|_F^2$, where $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times n}$ and $Z = [z_1, z_2, \ldots, z_n] \in \mathbb{R}^{p \times n}$. Let $\hat{Z}$ be the given converged sparse codes, then stationary points of the problem w.r.t the network weights (dictionary) follows $\hat{D} = X G^{-1} \hat{Z}^T$, where we denote $G := (\hat{Z}^T \hat{Z} + \omega I)^{-1}$

The dictionary interpolates the training data Given Theorem 5.1, each learned atom interpolates the training data, i.e.,

$$\hat{D}_j = X(G^{-1} w_j) = \sum_{k=1}^{n} (G^{-1} w_j)_k x^k \quad \text{(17)}$$

where $w_j = [\hat{z}_1^j, \hat{z}_2^j, \ldots, \hat{z}_n^j]^T \in \mathbb{R}^n$ is a vector containing the training code activity for dictionary atom $j$. Specifically, the importance of training image $x^k$ in learning dictionary atom $j$ is captured by the term $(G^{-1} w_j)_k$. This proves the dictionary lives in the spans of the

Figure 7: Fraction of dictionary atoms learned from $\{0, 1, 2, 3, 4\}$ MNIST.
training set. Given the small number of atoms compared to the training size, (17) shows that the dictionary summarizes the training examples. We trained the network on digits of \{0, 1, 2, 3, 4\} MNIST (Figure 7 shows a fraction of the most used learned atoms). Figure 8 visualizes dictionary atoms along with training images with the highest contribution (green) and the lowest contribution (red). In addition, we used (17) on the partial training data to reconstruct learned atoms (shown as Estimate). Next, we interpret the relation between a new data to the training data using representer point selection, similar to (Yeh et al., 2018).

We observe that the contribution of image \(\hat{x} = \tilde{K}z\hat{j}\) with the highest contribution (green) and the lowest contribution (red). In addition, we used (17) on the partial training data to reconstruct learned atoms (shown as Estimate). Next, we interpret the relation between a new data to the training data using representer point selection, similar to (Yeh et al., 2018).

Relation between new test image and training data For representation of a new data, we observe that the reconstruction of a new example \(x^j\) is a linear combination of all the training examples, i.e.,

\[
\hat{x}^j = \hat{D}z^j = X\beta^j = \sum_{k=1}^{n} \beta^j_k x^k
\]

where \(\hat{x}^j\) denotes reconstruction, \(z^j\) is the code estimate, \(\beta^j = G^{-1}\tilde{Z}^T\hat{z}^j \in \mathbb{R}^n\), and \(\beta^j_k = \sum_{a=1}^{n} G^{-1}_{ka}\langle z^a, \hat{z}^j \rangle\).

We observe that the contribution of image \(k\) into the reconstruction of the test image is a function of \(\beta^j_k\), and the energy of \(\beta^j_k\) itself depends on the whole training set, and \(G^{-1}\). (18) shows how each image is reconstructed as interpolation of the training images. Figure 9 shows this results, where images with high (green) \(\beta^j_k\) contribution are similar to the test image and those with low (red) \(\beta^j_k\) contribution are different.

In addition, we can evaluate the overall quality of the reconstruction by looking into \(\beta^j_k\) in (18). For example, we observed that for test MNIST, unnormalized \(\beta^j_k\) corresponding to high contributing training images is above 1. However, for resized-CIFAR, unnormalized \(\beta^j_k\) of high contributing training images are often half or an order of magnitude lower than the MNIST case. This informs us of a bad representation/reconstruction of CIFAR image by the trained network. From another perspective, we can write the new image as

\[
x^j = \hat{D}z^j = \sum_{k=1}^{n} (XG^{-1})_k \langle \hat{z}^k, \hat{z}^j \rangle
\]

i.e., the contribution of each training image for reconstruction is a function of their code similarity to the new image and properties of the Gram matrix of training set code similarities. Specifically, the relation rules the contribution of transformed image \(k\) (i.e., \((XG^{-1})_k\)) into reconstruction of the test image as a function of its code similarity \(\langle \hat{z}^k, \hat{z}^j \rangle\). In other words, (19) shows that training images with the highest code similarity to the representation of the new image have the highest contribution to its reconstruction. This interpretation is demonstrated in Figure 10. The training images with the highest code similarity (green) and the lowest
Figure 10: Contribution of images with code similarity into reconstruction of a new test image along with the histograms of the similarity of the test code to training codes from each class.

6 Conclusions

This paper studied dictionary learning and analyzed the dynamics of unrolled sparse coding networks through a provable unrolled dictionary learning (PUDLE) framework. First, we provided a theoretical analysis of the forward pass for code recovery and dictionary learning. We discussed the bias introduced by $\ell_1$-based sparse coding in the forward pass, and how this affects the dictionary estimate in the backward pass. Second, we showed strategies to mitigate the propagation of this code bias into the backward pass; this is achieved by modification of the training loss function. We demonstrated that this bias could be further reduced and eliminated by decaying the regularization parameter within the unrolled layers. Additionally, we provided sufficient conditions on the data distribution and network to guarantee stability of backpropagated gradient computations. In the absence of such conditions, we proposed a modification to the loss function that resolves the gradient explosion and allows stable learning. In an image denoising task, we showed PUDLE outperforms the NOODL sparse coding scheme (Rambhatla et al., 2018). Motivated by interpretability as a popular feature for unrolled networks, we derived a mathematical relation between the network weights (dictionary) and the training set. We proved that the network weights live in the span of the training set, and constructed a relation between predictions of new input examples and the training set. The latter allows the user to extract images from the training set that are similar/dissimilar to the input image in representation/reconstruction.

Acknowledgments

Demba Ba and Bahareh Tolooshams acknowledge the partial support of the ARO under grant number W911NF-16-1-0368. This is part of a collaboration between US DOD, UK MOD and UK Engineering and Physical Research Council (EPSRC) under the Multidisciplinary University Research Initiative. The authors would also like to thank the reviewers for their feedback, led to the improvement of the manuscript.

References


## A Appendix - proofs

### A.1 Notation

Bold-lower-case and upper-case letters refer to vectors $d$ and matrices $D$. We use $d_{(j)}$ to denote the $j^{th}$ element of the vector $d$, and $D_{(j)}$ is the $j^{th}$ column of the matrix $D$. We denote the code estimate at unrolled layer $t$ by $z_t$, $\lambda > 0$ is the regularization (sparsity-enforcing) parameter. $\sigma_{\text{max}}(D)$ is the maximum singular value of $D$. When taking the derivatives or norms w.r.t the matrix $D$, we assume that $D$ is vectorized. $\nabla_1 L_x(z, D)$ and $\nabla_2 L_x(z, D)$ are the first derivatives of the loss w.r.t $z$ and $D$, respectively. $\nabla_1^2 L_x(z, D)$ is the second derivative of the loss w.r.t $z$. $\nabla_2^2 L_x(z, D)$ is the derivative of $\nabla_1 L_x(z, D)$ w.r.t $D$. The support of $z$ is supp($z$) $\triangleq \{j : z_{(j)} \neq 0\}$.

### A.2 Basic definitions and Lemmas

We list four definitions used throughout the paper below.

**Definition A.1** ($\mu$-incoherence). $D$ is $\mu$-incoherent, i.e., for every pair $(i, j)$ of columns, $|\langle D_i, D_j \rangle| \leq \mu/\sqrt{m}$.

**Definition A.2** ($\delta, \kappa$-closeness). Dictionary $D$ is $\delta$-close to $D^*$, i.e., there is a permutation $\pi$ and sign flip operator $u$ such that $\forall i \| u(i) D_{(\pi(i))} - D_{(i)}^* \|_2 \leq \delta$. Additionally, $\| D - D^* \|_2 \leq \kappa \| D^* \|_2$.

**Definition A.3** (Lipschitz function). A function $f: \mathbb{R}^m \rightarrow \mathbb{R}^p$ is $L$-Lipschitz w.r.t a norm $\| \cdot \|$ if $\exists L > 0$ s.t. $\| f(a) - f(b) \| \leq L \| a - b \|$ $\forall a, b \in \mathbb{R}^m$.

**Definition A.4** (Lipschitz differentiable function). A twice differentiable function $f: \mathbb{R}^m \rightarrow \mathbb{R}^p$ is $L$-Lipschitz differentiable w.r.t a norm $\| \cdot \|$ iff $\exists L > 0$ s.t. $\| \nabla^2 f(a) \| \leq L \forall a \in \mathbb{R}^m$.

**Definition A.5** (Strong convexity). A twice differentiable function $f: \mathbb{R}^m \rightarrow \mathbb{R}^p$ is strongly convex if $\exists \mu > 0$ s.t. $\nabla^2 f(a) \succeq \mu I$. 


Definition A.6 (Norm of subgradient). For norms involving subgradients, we define $\|\partial h(z)\| := \max_{v \in \partial h(z)} \|v\|$.

In the proof of the theorems, we use the strong convexity of the reconstruction loss after support selection and the bounded property of the Lipschitz mapping stated below.

Lemma A.1 (Strong convexity of reconstruction loss). Given the support selection (Proposition 4.1), $D_S^T D_S$ is full-rank. Thus, $\forall t > B$, $\mathcal{L}_x(z_t, D) = \mathcal{L}_x(z_t, D_S)$ is strongly convex (Definition A.5) in $z$.

Lemma A.2 (Lipschitz mapping). Given the recursion $z_{t+1} = \Phi(z_t) = \mathcal{P}_\alpha(z_t - \alpha \nabla_1 \mathcal{L}_x(z_t, D))$, from Lemma A.1, there exist $B > 0$ such that $\mathcal{L}_x(z_t, D)$ is $\mu$-strongly convex $\forall t > B$. Hence, using Lemma A.4,

$$\|\nabla \Phi(z_t, D)\|_2 = \|(I - \alpha \nabla_1 \mathcal{L}_x(z_t, D))\partial \mathcal{P}_\alpha(z_t)\|_2 \leq \rho$$

where $\rho := c_{\text{prox}}(1 - \alpha \mu) < 1$.

One key term used in the proofs, is that $0 \in \nabla_1 \mathcal{L}_x(z, \hat{D}) + \partial h(z)$ which is followed by the lasso optimality, i.e.,

Lemma A.3 (Lasso optimality). Lasso Karush-Kuhn-Tucker (KKT) optimality conditions are

$$\hat{z} \in \mathbb{R}^p$$

One key term used in the proofs, is that $0 \in \nabla_1 \mathcal{L}_x(z, \hat{D}) + \partial h(z)$ which is followed by the lasso optimality, i.e.,

A.3 Forward pass proof details

Given the $\mu$-incoherence of $D^*$, and current dictionary closeness of $\delta_t$, we re-state Lemma 3.1 and prove it below. It shows that the current dictionary is $\mu_t$-close to $D^*$.

Lemma 3.1 ($\mu_t$-incoherent). $D^{(i)}$ is $\mu_t$-incoherent where $\mu_t = \mu + 2\sqrt{m}\delta_t$.

Proof.

$$\langle D^{(i)}_i, D^{(i)}_j \rangle = \langle D^*_i, D^*_j \rangle - \langle D^*_i - D^{(i)}_i, D^*_j \rangle - \langle D^{(i)}_i - D^*_i, D^*_j \rangle$$

$$\|D^{(i)}_i - D^*_i\| \leq \mu_t \sqrt{m} + \|D^*_i - D^{(i)}_i\|_2\|D^*_j\|_2 + \|D^{(i)}_i\|_2\|D^*_j - D^{(i)}_j\|_2 \leq \mu_t \sqrt{m} + 2\delta_t$$

We re-state and proof the forward pass support recovery (Theorem 4.1). This shows that given proper initialization and under mild conditions, the support of the true code $z^*$ is recovered with high probability in one iteration of the encoder.

Theorem 4.1 (Forward pass support recovery). Given Assumptions 3.3 and 3.4, suppose $D^{(i)}$ is $\delta_t = \mathcal{O}^\ast(1/\sqrt{\log p})$ close to $D^*$. If $s = \mathcal{O}^\ast(\sqrt{m}/\mu \log m)$, and $\mu = \mathcal{O}(\log m)$, then with probability of at least $1 - \epsilon_{\text{supp-rec}}$, the choice of $\lambda_0 = C_{\text{min}}/4$ recovers the support of the code $z^*$ in one encoder iteration, i.e.,

$$\text{sign}(S_{\alpha, \lambda_0}(\alpha D^{(i)} T x)) = \text{sign}(z^*)$$

Proof. The code estimate after one iteration is $z_1 = \mathcal{P}_\alpha(\alpha D^{(i)} T x) = \text{sign}(D^{(i)} T x) \text{ReLU}(\alpha(D^{(i)} T D^* z^*) - \lambda_0)$. We focus on the positive entries. The analysis for negative entries is similar. Writing the relation for $i$-th entry,

$$z_{1,(i)} = \text{sign}(D^{(i)} T x) \text{ReLU}(\alpha(\sum_{j \in S^*} \langle D^{(i)}_i, D^*_j \rangle z^*_j - \lambda_0))$$

$$= \text{ReLU}(\alpha(\langle D^{(i)}_i, D^*_i \rangle z^*_i + \sum_{j \in S^* \setminus \{i\}} \langle D^{(i)}_i, D^*_j \rangle z^*_j - \lambda_0))$$

We focus on the term inside ReLU and discard $\alpha$, shared by all terms. We shows that under proper choice of $\lambda_0$, $\langle D^{(i)}_i, D^*_i \rangle z^*_i$ is greater than $\lambda_0$ and $v_i = \sum_{j \in S^* \setminus \{i\}} \langle D^{(i)}_i, D^*_j \rangle z^*_j$ is small with respect to $\lambda_0$, hence
getting cancelled by ReLU. The small value of \( v_{i} \), compared to \( \langle D_{i}^{(l)}, D_{i}^{*}\rangle z_{i}^{*} \), results in \( \text{sign}(D^{(l)T}x) \) be equal to the sign\( (D^{(l)T}D^*)z^* \) which is equal to the sign of \( z^* \).

Given the current dictionary distance \( \|D_{i}^{(l)} - D^*\|_2 \leq \delta_l \), we can find a lower bound on \( \langle D_{i}^{(l)}, D_{i}^{*}\rangle z_{i}^{*} \) as follows
\[
\langle D_{i}^{(l)}, D_{i}^{*}\rangle = \frac{1}{2}(\|D_{i}^{(l)}\|_2^2 + \|D_{i}^{*}\|_2^2 - \|D_{i}^{(l)} - D^*\|_2^2) = 1 - \frac{1}{2}\|D_{i}^{(l)} - D^*\|_2^2
\]
\[
|\langle D_{i}^{(l)}, D_{i}^{*}\rangle| \geq 1 - \delta_l^2/2
\]

Hence, for \( i \in S^* \)
\[
|\langle D_{i}^{(l)}, D_{i}^{*}\rangle z_{i}^{*} | \geq (1 - \delta_l^2/2)C_{\text{min}}
\]
otherwise, it is 0. Given, \( \text{var}(z_{i}^{*}) = 1 \) for \( i \in S^* \), we find an upper bound on the variance \( v_{i} \) of as follows
\[
\text{var}(v_{i}) = \sum_{j \in S^* \setminus \{i\}} \langle D_{i}^{(l)}, D_{j}^{*}\rangle^2 = \sum_{j \in S^* \setminus \{i\}} (\langle D_{i}^{*}, D_{j}^{*}\rangle + \langle D_{i}^{(l)} - D^*, D_{j}^{*}\rangle)^2 \leq \sum_{j \in S^* \setminus \{i\}} 2(\langle D_{i}^{*}, D_{j}^{*}\rangle^2 + \langle D_{i}^{(l)} - D^*, D_{j}^{*}\rangle^2) \leq \sum_{j \in S^* \setminus \{i\}} (2\mu^2/m) + 2\|\langle D_{i}^{(l)} - D^*\rangle^T D^*_{S^* \setminus \{i\}}\|_2^2 \leq 2(2\mu^2/m + 4\delta_l^2) = O^*(\delta_l^2)
\]
where we used the Gershgorin Circle Theorem for the bound \( \|D^*_{S^* \setminus \{i\}}\|_2 \leq 2 \). With the sub-Gaussian assumption on the coefficients \( z^* \), we get the following using Chernoff bound concerning \( v_{i} \).
\[
P(|v_{i}| \geq \frac{C_{\text{min}}}{4}) \leq 2\exp\left(-\frac{C_{\text{min}}^2}{4\delta_l^2} + \frac{16\delta_l^2}{\delta_l^2}\right) = 2\exp\left(-\frac{C_{\text{min}}^2}{O^*(\delta_l^2)}\right)
\]
Taking a union bound over all indices \( i \in [1, p] \) will result in
\[
P(\max_i |v_{i}| \geq \frac{C_{\text{min}}}{4}) \leq 2p\exp\left(-\frac{C_{\text{min}}^2}{O^*(\delta_l^2)}\right) := \varepsilon_{\text{supp-rec}}^{(l)}
\]
Hence, we can set \( \lambda_0 = C_{\text{min}}/2 \).

We re-state and prove the forward pass support preservation (Theorem 4.2).

**Theorem 4.2 (Forward pass support preservation).** Given Assumptions 3.3 and 3.4, suppose \( D^{(l)} \) is \( \delta_l = O^*(1/\sqrt{\log p}) \) close to \( D^* \). If \( s = O^*(\sqrt{m}/\mu \log m) \), \( \mu = O(\log m) \), and the regularizer and step size are chosen such that \( \lambda_l^{(l)} = \frac{\mu}{\sqrt{m}}\|z^* - z_t\|_1 + \alpha_l = O(\frac{\log m}{\sqrt{m}}) \) and \( \alpha_l^{(l)} \leq 1 - 2\frac{\lambda_l^{(l)} - (1 - \frac{1}{\delta_l^2})C_{\text{min}}}{\lambda_l^{(l)} - 1} \), then with probability of at least \( 1 - \varepsilon_{\text{supp-pres}}^{(l)} \), the support, recovered at the first iteration, is preserved through the encoder iterations. We have \( \alpha_l = O(\sqrt{\delta_l}) \) and \( \varepsilon_{\text{supp-pres}}^{(l)} := \varepsilon_{\text{supp-rec}}^{(l)} + \varepsilon_{\gamma}^{(l)} = 2p\exp\left(-\frac{C_{\text{min}}^2}{O^*(\delta_l^2)}\right) + 2s\exp\left(-\frac{C_{\text{min}}^2}{O^*(\delta_l^2)}\right) \).

**Proof.** Given current dictionary \( D^{(l)} \), in each iteration of the forward pass, we have \( z_{t+1} = P_{\alpha_l}(z_t + \alpha D^T(z^* - Dz_t)) \). We focus on the entries that are non-negative. Then procedure for negative code entries is similar. We follow similar steps as in (Rambhatla et al., 2018). We get
\[
z_{t+1,(j)} = \text{ReLU}(\langle I - \alpha D^{(l)T}D^{(l)}(j,:), z_t + \alpha(D^{(l)T}D^*)_{(j,:)}, z^* - \alpha \lambda_{t,j}^{(l)}\rangle) \\
= \text{ReLU}(\langle I - \alpha D^{(l)T}D^{(l)}(j,:), z_t + \alpha((D^{(l)} - D^*)^T D^*)_{(j,:)}, z^* + \alpha(D^*T D^*)_{(j,:)}, z^* - \alpha \lambda_{t,j}^{(l)}\rangle) \\
= \text{ReLU}(\langle I - \alpha z_{t,(j)} - \alpha \sum_{i \neq j} \langle D_{i}^{(l)}, D_{i}^{(l)}(j,:), z_t(i) + \alpha((D_{i}^{(l)}(j,:) - D^*_{(j,:)})D^*)_{(j,:)}, z^*(i) - \alpha \lambda_{i,j}^{(l)}\rangle) \\
+ \alpha \langle D_{j}^{(l)} - D_{j}^{*}, D_{j}^{*}, z_{i}^{*}(i) + \alpha z_{j}(j) + \alpha \sum_{i \neq j} \langle D_{j}^{(l)}, D_{j}^{*}, z_{i}^{*}(i) - \alpha \lambda_{i,j}^{(l)}\rangle \\
= \text{ReLU}(\langle I - \alpha z_{t,(j)} + \alpha(1 - \beta_{j}^{(l)} z_{j}(j) + \alpha \lambda_{i,j}^{(l)} - \alpha \lambda_{i,j}^{(l)}\rangle)
\]
where $\beta_j^{(l)} = \langle D_j^* - D_j^{(l)}, D_j^* \rangle$, and $\eta_{t,j}^{(l)} = -\sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle z_{t,(i)} + \langle \langle D_j^{(l)} - D_j^*, D_j^* \rangle + \langle D_j^*, D_i^* \rangle \rangle z_{t,(i)}$. With $\|D_j^{(l)} - D_j^*\|_2 \leq \delta_1$, $\beta_j^{(l)}$ can be bounded as follows

$$\beta_j^{(l)} = \langle D_j^* - D_j^{(l)}, D_j^* \rangle \leq \delta_1^2 / 2 \quad (30)$$

where we used the relation $\|D_j^{(l)} - D_j^*\|_2^2 = 2(1 - \langle D_j^{(l)}, D_j^* \rangle)$. We re-write $\eta_{t,j}^{(l)}$ below

$$\eta_{t,j}^{(l)} = -\sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle z_{t,(i)} + \langle \langle D_j^{(l)} - D_j^*, D_j^* \rangle + \langle D_j^*, D_i^* \rangle \rangle z_{t,(i)}$$

$$= -\sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle z_{t,(i)} + \sum_{i \neq j} \langle \langle D_j^{(l)} - D_j^*, D_j^* \rangle + \langle D_j^*, D_i^* \rangle \rangle z_{t,(i)}$$

$$= \sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle (z_{t,(i)} - z_{t,(i)}) + \sum_{i \neq j} \langle \langle D_j^{(l)} - D_j^*, D_j^* \rangle + \langle D_j^*, D_i^* \rangle \rangle z_{t,(i)}$$

$$= \sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle (z_{t,(i)} - z_{t,(i)}) + \sum_{i \neq j} \langle D_j^{(l)}, D_i^* - D_i^{(l)} \rangle z_{t,(i)}$$

$$= \sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle (z_{t,(i)} - z_{t,(i)}) + \gamma_j^{(l)} \quad (31)$$

where $\gamma_j^{(l)} = \sum_{i \neq j} \langle D_j^{(l)}, D_i^* - D_i^{(l)} \rangle z_{t,(i)}$. Given the sub-Gaussian entries of the code $z^*$, we provide a bound on the variance of $\gamma_j^{(l)}$ below:

$$\text{var}(\gamma_j^{(l)}) = \sum_{i \neq j} \langle D_j^{(l)}, D_i^* - D_i^{(l)} \rangle^2 \leq s\delta_1^2 \quad (32)$$

Now, using Chernoff bound on the sub-Gaussian code entries, we get

$$P(\{ |\gamma_j^{(l)} | > a \}) \leq 2 \exp \left( \frac{-a^2}{2s\delta_1^2} \right) \quad (33)$$

To bound all the terms in the support, for $j \in S^*$, we have

$$P(\max_j |\gamma_j^{(l)} | > a_{\gamma}) \leq \epsilon_{\gamma}^{(l)} \quad (34)$$

where $\epsilon_{\gamma}^{(l)} = 2s \exp \left( \frac{-a_{\gamma}^2}{2s\delta_1^2} \right)$. Let $\alpha_{\gamma} = \Theta(\sqrt{s}\delta_1)$, then $\epsilon_{\gamma}^{(l)} = 2s \exp \left( \frac{-1}{\Theta(\delta_1)} \right)$. The above analysis states that with probability of at least $1 - \epsilon_{\gamma}^{(l)}$, $|\gamma_j^{(l)} | \leq a_{\gamma} = \Theta(\sqrt{s}\delta_1)$. Next, we write the recursion for when the support is identified (see Theorem 4.1). For the code at iteration $T$, we have

$$z_{T,(j)} = (1 - \alpha)^T z_{0,(j)} + z_{j}^{(l)} \sum_{t=1}^{T} \alpha(1 - \beta_j^{(l)})(1 - \alpha)^{T-t} + \sum_{t=1}^{T} \alpha(\eta_{t-1,j}^{(l)} - \lambda_{t-1,j}^{(l)})(1 - \alpha)^{T-t}$$

$$= (1 - \alpha)^T z_{0,(j)} + z_{j}^{(l)}(1 - \beta_j^{(l)})(1 - (1 - \alpha)^{T}) + \sum_{t=1}^{T} \alpha(\eta_{t-1,j}^{(l)} - \lambda_{t-1,j}^{(l)})(1 - \alpha)^{T-t} \quad (35)$$

$$= z_{j}^{*}(1 - \beta_j^{(l)}) + (1 - \alpha)^T (z_{0,(j)} - z_{j}^{*}(1 - \beta_j^{(l)})) + \sum_{t=1}^{T} \alpha(\eta_{t-1,j}^{(l)} - \lambda_{t-1,j}^{(l)})(1 - \alpha)^{T-t}$$

$$= z_{j}^{*}(1 - \beta_j^{(l)}) + \zeta_{T,j}^{(l)}$$

where $\zeta_{T,j}^{(l)} = (1 - \alpha)^T (z_{0,(j)} - z_{j}^{*}(1 - \beta_j^{(l)})) = \sum_{t=1}^{T} \alpha(\eta_{t-1,j}^{(l)} - \lambda_{t-1,j}^{(l)})(1 - \alpha)^{T-t}$. With the support correctly identified at iteration $t - 1$, we show that the support is preserved at iteration $t$. With $\|z^* - z_t\|_1 = \Theta(s)$, for each $j \in S^*$, we have

$$\eta_{t,j}^{(l)} = \sum_{i \neq j} \langle D_j^{(l)}, D_i^{(l)} \rangle (z_{j}^{*}(1 - \beta_j^{(l)})) + \gamma_j^{(l)} \leq \frac{\mu_1}{\sqrt{m}} \|z^* - z_t\|_1 + a_{\gamma} = \Theta(\frac{s\log m}{\sqrt{m}}) \quad (36)$$
We make sure the regularizer is chosen such that
\[ \lambda_t \geq \frac{\mu}{\sqrt{m}} \|z^* - z_t\|_1 + a, \]  
\[ (37) \]

We see that the larger the code error and coherence between the columns of the current dictionary, the larger \( \lambda_t \) should be. This is to suppress the noise component in the code recursion and make sure no false support is introduced. Furthermore, we want \( \lambda_t \) to be lower than half of the signal component, i.e.,

\[ \alpha \lambda_t \leq \frac{1 - \alpha}{2} z_{t, (j)} + \frac{\alpha}{2} (1 - \beta^{(l)}) z^*_j, \forall j \in S^* \]
\[ \alpha \lambda_t \leq \frac{1 - \alpha}{2} z_{t, \min} + \frac{\alpha}{2} (1 - \frac{\delta^2}{2}) C_{\min} \]
\[ (38) \]

where \( z_{t, \min} = \min_t z_{t, (j)} \). We further shrink the upper bound, given the code from previous iteration \( t-1 \) (i.e., \( \alpha \lambda_{t-1} \leq z_{t, \min} \)). Hence, we want the regularizer to follow

\[ \alpha \lambda_t \leq \frac{1 - \alpha}{2} \alpha \lambda_{t-1} + \frac{\alpha}{2} (1 - \frac{\delta^2}{2}) C_{\min} \]
\[ \lambda_t \leq \frac{1 - \alpha}{2} \lambda_{t-1} + \frac{1}{2} (1 - \frac{\delta^2}{2}) C_{\min} \]
\[ (39) \]

This condition is to make sure the identified supports are not killed in the recursion. We use the condition to set the step size \( \alpha \). We get

\[ \alpha \leq 1 - \frac{2 \lambda_t - (1 - \frac{\delta^2}{2}) C_{\min}}{\lambda_{t-1}} \]
\[ (40) \]

Hence, \( \lambda_t = \Omega(\frac{1}{m} \log m) \) and \( \alpha \) should be chosen sufficiently small such that the condition above is met. We denote \( \epsilon^{(l)}_{\text{supp-pres}} := \epsilon^{(l)}_{\text{supp-rec}} + \epsilon^{(l)} = 2p \exp \left( \frac{C_{\min}^2}{\Omega(\delta^2)} \right) + 2s \exp \left( \frac{1}{\Omega(\delta^2)} \right) \). Hence, with probability of at least \( 1 - \epsilon^{(l)}_{\text{supp-pres}} \), the support, recovered at the first iteration, is preserved through the encoder iterations. \[ \blacksquare \]

**Theorem 4.1** and **Theorem 4.2** allow to achieve linear convergence in the forward pass right after the first encoder iteration, i.e., \( B = 1 \). With support recovery at first iteration and its preservation, we now re-state the forward pass code convergence (**Theorem 4.3**).

**Theorem 4.3** (Local forward pass code convergence). Given the encoder \( z_{t+1} = \Phi(z_t, D) \), **Assumption 3.6, Lemmas 3.2, A.1 and A.2**, then \( \exists \rho < 1, B > 0 \text{ s.t. } ||z_t - \hat{z}||_2 \leq \mathcal{O}(\rho^t) \forall t > B \), where \( \hat{z} \) is the unique minimizer of lasso (1). Furthermore, given **Theorem 4.1** and **Theorem 4.2**, \( B = 1 \).

**Proof.** Given the support selection at iteration \( B \), from **Lemma A.1**, we have \( \nabla^2_{z_1} L_x(z_t, D) \geq \mu I \) restricted to the support for \( t > B \). Then, from **Lemma A.2**, we get

\[ ||\nabla \Phi(z_t, D)||_2 = ||(I - \alpha \nabla^2_{z_1} L_x(z_t, D)) \partial \mathcal{P}_{\alpha \lambda}(z_t)||_2 \leq \rho \]
\[ \text{where } \rho \triangleq c_{\text{prox}} (1 - \alpha \mu) < 1 \]. Hence, using fixed-point property (**Lemma 3.2**)

\[ \exists B > 0, \text{ s.t. } ||z_{t+1} - \hat{z}||_2 = ||\Phi(z_t) - \Phi(\hat{z})||_2 \leq \rho ||z_t - \hat{z}||_2 \forall t > B \]

where \( \hat{z} = \arg \min_x f_x(z, D) \). Unrolling the recursion,

\[ ||z_t - \hat{z}||_2 \leq \rho^{t-B} ||z_B - \hat{z}||_2. \]
\[ \blacksquare \]

**Theorem 4.4** (Global forward pass code error with variable \( \lambda_t \)). Given **Assumptions 3.3 and 3.4**, suppose \( D^{(l)} \) is \( \mu_1 \)-incoherent and \( \delta_1 = \mathcal{O}^*(1/\log p) \) close to \( D^* \). If \( s = \mathcal{O}^*(\sqrt{m}/\mu \log m) \), \( \mu = \mathcal{O}(\log m) \), and the regularizer
and step size are chosen such that $\lambda_{t}^{(l)} = \frac{\mu l}{\sqrt{m}} \| z^* - z_t \|_1 + a = \Omega \left( \frac{\log m}{\sqrt{m}} \right)$ and $\alpha^{(l)} \leq 1 - \frac{2 \lambda_{t-1}^{(l)} - \frac{\mu l}{\sqrt{m}}}{\lambda_{t-1}^{(l)}}$, then with probability of at least $1 - \epsilon_{t-1}^{(l)}$, for $j \in \text{supp}(z^*)$, the code coefficient error is

$$
|z_{t,(j)}^{(l)} - z_j^*| \leq \mathcal{O}(\sqrt{s} \| D_j^{(l)} - D_j^* \|_2 + \epsilon_t^{(l)\text{unroll}})
$$

(8)

and

$$
z_{t,(j)} = z_{t,j}^*(1 - \beta_j^{(l)}) + \xi_{t,j}^{(l)}
$$

(9)

where $\epsilon_{t,i}^{(l)\text{unroll}} := 2(s-1)\alpha \frac{\mu l}{\sqrt{m}} \max_i |z_{0,i}^{(l)} - z_i^*| \delta_{a,1} \delta_{t-1} + |z_{t,i}^{(l)} - z_i^*| \delta_{a,1} \delta_{t} \delta_{a,1,t} \delta_{t} := (1 - \alpha + 2\alpha \frac{\mu l}{\sqrt{m}})^t \eta_{t,i}^{(l)}$,

$$
|\xi_{t,j}^{(l)}| = \mathcal{O}(a) \text{ with } a = \mathcal{O}(\sqrt{s} \delta_{t}), \beta_j^{(l)} = \langle D_j - D_j^{(l)}, D_j^* \rangle \leq \frac{\delta^2_j}{2} \text{ and } \epsilon_{t-1}^{(l)\text{unroll}} := \epsilon_{t-1}^{(l)\text{rec}} + \epsilon_{t-1}^{(l)} = 2p \exp \left( -\frac{C_{\text{supp}}^2}{2(t_j)} \right) + 2s \exp \left( -\frac{C_{\text{supp}}}{2(t_j)} \right). \quad \text{With appropriately large } t, |z_{t,(j)}^{(l)} - z_j^*| = \mathcal{O}(\sqrt{s} \| D_j^{(l)} - D_j^* \|_2).
$$

Proof. We define $\eta_{t,i}^{(l)} := \sum_{j \neq i} \| D_j^{(l)} - D_i^{(l)} \| z_{t,i}^* - z_{t,j}^* \| + \gamma_j^{(l)}$ and upper bound it as

$$
\eta_{t,i}^{(l)} \leq \frac{\mu l}{\sqrt{m}} \sum_{j \neq i} E_{t,i} + \gamma_j^{(l)}
$$

(41)

where $E_{t,i} := |z_{t,i}^* - z_{t,(i)}|$. Given (21), we re-write the code recursion

$$
z_{t+1,(j)} = P_{\alpha^{(l)}} \left( (1 - \alpha) z_{t,(j)} + \alpha (1 - \beta_j^{(l)}) z_j^* + \alpha \eta_{t,i}^{(l)} \right) \in (1 - \alpha) z_{t+1,(j)} + \alpha (1 - \beta_j^{(l)}) z_j^* + \alpha (\eta_{t,i}^{(l)} + \lambda_{t,j}^{(l)}) |z_{t+1,(j)}|
$$

Opening up the recursion, we get

$$
E_{t+1,j} = E_{0,j} \prod_{q=0}^{t} (1 - \alpha) + \beta_j^{(l)} |z_j^*| \sum_{a=1}^{t+1} \alpha \prod_{q=0}^{t+1} (1 - \alpha) + \sum_{a=1}^{t+1} \alpha (\eta_{a-1,j}^{(l)} + \lambda_{a-1,j}^{(l)}) \prod_{q=a}^{t+1} (1 - \alpha)
$$

(42)

where we define $\prod_{q=a}^{t} (1 - \alpha) = 1$. Using the upper bound from (41) and $\lambda_{t,j}^{(l)}$ from Theorem 4.2, we get

$$
E_{t+1,j} \leq v_{t+1,j} + 2 \frac{\mu l}{\sqrt{m}} \alpha \sum_{a=1}^{t+1} E_{a-1,i} \prod_{q=a}^{t+1} (1 - \alpha)
$$

(43)

where $v_{t+1,j} = E_{0,j} \prod_{q=0}^{t} (1 - \alpha) + (\beta_j^{(l)} |z_j^*| + 2 \gamma_j^{(l)} \sum_{a=1}^{t+1} \alpha \prod_{q=a}^{t+1} (1 - \alpha)$. We now derive the general upper bound on $E_{t+1,j}$ as follows

$$
E_{1,i_1} \leq v_{1,i_1} + 2 \alpha \frac{\mu l}{\sqrt{m}} \sum_{i_2 \neq i_1} E_{0,i_2}
$$

(45)

For $E_{2,i_1}$, we have

$$
E_{2,i_1} \leq v_{2,i_1} + 2 \alpha \frac{\mu l}{\sqrt{m}} \left( \sum_{i_2 \neq i_1} E_{1,i_2} + \sum_{i_2 \neq i_1} E_{0,i_2} (1 - \alpha) \right)
$$

(46)

Substituting $E_{1,i_1}$,

$$
E_{2,i_1} \leq v_{2,i_1} + 2 \alpha \frac{\mu l}{\sqrt{m}} \left( \sum_{i_2 \neq i_1} (v_{1,i_2} + 2 \alpha \frac{\mu l}{\sqrt{m}} \sum_{i_3 \neq i_2} E_{0,i_3}) + \sum_{i_2 \neq i_1} E_{0,i_2} (1 - \alpha) \right)
$$

(47)

For $E_{3,i_1}$, we have

$$
E_{3,i_1} \leq v_{3,i_1} + 2 \alpha \frac{\mu l}{\sqrt{m}} \sum_{a=1}^{3} \sum_{i_2 \neq i_1} E_{0,i_2} (1 - \alpha)^{3-a}
$$

(48)
Unrolling the recursion,

$$E_{3,i_t} \leq v_{3,i_t} + 2\alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_2 \neq i_1} v_{2,i_2} + 2\alpha \frac{\mu_t}{\sqrt{m}} \left( (1-\alpha) \sum_{i_2 \neq i_1} v_{1,i_2} + 2\alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_3 \neq i_2,i_1} v_{1,i_3} \right)$$

$$+ 2\alpha \frac{\mu_t}{\sqrt{m}} \left( (1-\alpha)^2 \sum_{i_2 \neq i_1} E_{0,i_2} + 2(1-\alpha)(2\alpha \frac{\mu_t}{\sqrt{m}})^2 \sum_{i_3 \neq i_2,i_1} E_{0,i_3} + (2\alpha \frac{\mu_t}{\sqrt{m}})^2 \sum_{i_4 \neq i_3,i_2,i_1} E_{0,i_4} \right)$$

(49)

Given above, we can write up the relation as

$$E_{3,i_t} \leq v_{3,i_t} + 2\alpha \frac{\mu_t}{\sqrt{m}} \left( (s-1)v_{2,i} + v_{1,i}((1-\alpha)(s-1) + 2\alpha \frac{\mu_t}{\sqrt{m}}(s-2)) \right)$$

$$+ 2\alpha \frac{\mu_t}{\sqrt{m}} E_{0,i} \left( (s-1)(1-\alpha)^2 + 2(1-\alpha)(2\alpha \frac{\mu_t}{\sqrt{m}})(s-2) + (2\alpha \frac{\mu_t}{\sqrt{m}})^2(s-3) \right)$$

$$\leq v_{3,i_t} + (s-1)2\alpha \frac{\mu_t}{\sqrt{m}} \left( v_{2,i} + v_{1,i}((1-\alpha) + 2\alpha \frac{\mu_t}{\sqrt{m}}) \right)$$

$$+ 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,i} \left( (1-\alpha)^3 + 3(1-\alpha)(2\alpha \frac{\mu_t}{\sqrt{m}})^2 + 3(1-\alpha)^2(2\alpha \frac{\mu_t}{\sqrt{m}}) + (2\alpha \frac{\mu_t}{\sqrt{m}})^3 \right)$$

(50)

Following similar steps for $E_{4,i_t}$, we get,

$$E_{4,i_t} \leq v_{4,i_t} + (s-1)2\alpha \frac{\mu_t}{\sqrt{m}} \left( v_{3,i} + v_{2,i}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}}) + v_{1,i}((1-\alpha)^2 + 2(1-\alpha)(2\alpha \frac{\mu_t}{\sqrt{m}}) + (2\alpha \frac{\mu_t}{\sqrt{m}})^2) \right)$$

$$+ 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,i} \left( (1-\alpha)^3 + 3(1-\alpha)(2\alpha \frac{\mu_t}{\sqrt{m}})^2 + 3(1-\alpha)^2(2\alpha \frac{\mu_t}{\sqrt{m}}) + (2\alpha \frac{\mu_t}{\sqrt{m}})^3 \right)$$

(51)

This leads to the term

$$E_{4,i_t} \leq v_{4,i_t} + (s-1)2\alpha \frac{\mu_t}{\sqrt{m}} \left( v_{3,i} + v_{2,i}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}}) + v_{1,i}((1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^2 \right)$$

$$+ 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,i}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^3$$

(52)

Hence, the general term for code error at $t$ layer is

$$E_{t+1,j} \leq v_{t+1,j} + 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} \sum_{a=1}^{t} v_{a,\max}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a} + 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,\max}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t}$$

(53)

where for $j$ in the support, we define the upper bounds $v_{a,j} \leq v_{a,\max}$ and $E_{0,j} \leq E_{0,\max}$. Next, we define $(1-\alpha)^t \leq \delta_{a,t} := (1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^t$, and use it to find an upper bound on the expression $\sum_{a=1}^{t} v_{a,\max}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a}$. We have

$$v_{t,j} = E_{0,j}(1-\alpha)^t + (\beta_{j}^{(t)} | z_{j}^* | \sum_{k=1}^{t} \alpha(1-\alpha)^{t-k+1}$$

(54)

We bound the expression

$$\sum_{a=1}^{t} v_{a,j}(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a} \leq \sum_{a=1}^{t} (E_{0,j}(1-\alpha)^a + (\beta_{j}^{(t)} | z_{j}^* | + 2\gamma_{j}^{(t)}) \sum_{k=1}^{a} \alpha(1-\alpha)^{a-k+1})(1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a}$$

$$\leq E_{0,j} t \delta_{a,t} + (\beta_{j}^{(t)} | z_{j}^* | + 2\gamma_{j}^{(t)}) \sum_{a=1}^{t} (1-\alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a} \sum_{k=1}^{a} \alpha(1-\alpha)^{a-k+1}$$

(55)
Using sum of geometric series, we write $\sum_{a=1}^{t} (1 - \alpha + 2a \frac{\mu \sqrt{m}}{\sqrt{t}})^{t-a} = \frac{1-(1-\alpha+2a \frac{\mu \sqrt{m}}{\sqrt{t}})^t}{\alpha - 2a \frac{\mu \sqrt{m}}{\sqrt{t}}} \leq \frac{1}{\alpha(1-\frac{\mu}{\sqrt{m}})}$. Hence, using (30) and (34), with probability of at least $1 - \epsilon_{\gamma}^{(l)}$, we have

$$
\sum_{a=1}^{t} v_{a,\text{max}} (1 - \alpha + 2a \frac{\mu \sqrt{m}}{\sqrt{t}})^{t-a} \leq E_{0,\text{max}} t \delta_{a,t} + \frac{1}{\alpha(1-2 \frac{\mu \sqrt{m}}{\sqrt{t}})} (\frac{\beta_{\gamma}^{(l)}}{2} |z_{\text{max}}^{(l)}| + 2a_{\gamma})
$$

(56)

Hence, we bound the code error on the coefficients as following

$$
E_{t+1,j} \leq v_{t+1,j} + 2(s-1) \frac{\mu \sqrt{m}}{\sqrt{t}} \left( \frac{1}{1 - \frac{\mu \sqrt{m}}{\sqrt{t}}} \right) \left( \frac{\beta_{\gamma}^{(l)}}{2} |z_{\text{max}}^{(l)}| + 2a_{\gamma} \right) + 2(t+1)(s-1) \alpha \frac{\mu \sqrt{m}}{\sqrt{t}} E_{0,\text{max}} \delta_{a,t}
$$

(57)

Next, we further simplify the first term

$$
v_{t+1,j} = E_{0,j} (1 - \alpha)^{t+1} + (\beta_{\gamma}^{(l)} |z_{(j)}^{(l)}| + 2\gamma_{\gamma}^{(l)}) \sum_{k=1}^{t+1} \alpha (1 - \alpha)^{t-k+1} \leq E_{0,j} \delta_{a,t} + \frac{\beta_{\gamma}^{(l)}}{2} |z_{\text{max}}^{(l)}| + 2a_{\gamma}
$$

(58)

Substituting the above upper bound into the upper bound for $E_{t+1,j}$, we get

$$
E_{t+1,j} \leq E_{0,j} \delta_{a,t} + (1 + 2(s-1)\kappa_{l}) \left( \frac{\beta_{\gamma}^{(l)}}{2} |z_{\text{max}}^{(l)}| + 2a_{\gamma} \right) + 2(t+1)(s-1) \alpha \frac{\mu \sqrt{m}}{\sqrt{t}} E_{0,\text{max}} \delta_{a,t}
$$

(59)

where $\kappa_{l} := \frac{\mu \sqrt{m}}{\sqrt{t}} \left( \frac{1}{1 - \frac{\mu \sqrt{m}}{\sqrt{t}}} \right)$. Given $s = \mathcal{O}^{*}(\sqrt{m}/\mu \log m)$, we have $(s-1)\kappa_{l} < 1$. Hence, with probability of at least $1 - \epsilon_{\gamma}^{(l)}$, we have

$$
E_{t,j} \leq \mathcal{O}(a_{\gamma}) + 2(s-1) t a \frac{\mu \sqrt{m}}{\sqrt{t}} E_{0,\text{max}} \delta_{a,t-1} + E_{0,j} \delta_{a,t}
$$

(60)

$$
|z_{t,(j)}^{(l)} - z_{(j)}^{(l)}| \leq \mathcal{O}(\sqrt{s} \|D_{j}^{(l)} - D_{j}^{*}\|_{2} + \epsilon_{t}^{(l)\text{unroll}})
$$

where $\epsilon_{t}^{(l)\text{unroll}} := 2(s-1) t a \frac{\mu \sqrt{m}}{\sqrt{t}} \max_{j} |z_{(j)}^{(l)} - z_{0,(j)}^{(l)}| \delta_{a,t-1} + |z_{0,(j)}^{(l)} - z_{(j)}^{(l)}| \delta_{a,t}$. With appropriately large unrolled layer $t$, $\epsilon_{t}^{(l)\text{unroll}} \approx 0$. Hence, for the code error on non-zero coefficients, we get

$$
|z_{t,(j)}^{(l)} - z_{(j)}^{(l)}| = \mathcal{O}(\sqrt{s} \|D_{j}^{(l)} - D_{j}^{*}\|_{2})
$$

(61)

for large enough $t$. Now, we try to prove the relation for $z_{T,(j)}$. For shrinkage, we re-write (35)

$$
z_{T,(j)} \in z_{(j)}^{*}(1 - \beta_{\gamma}^{(l)}) + \epsilon_{T,(j)}^{(l)}
$$

(62)

where $\epsilon_{T,(j)}^{(l)} = \kappa_{T,j}^{(l)} + \sum_{t=1}^{T} \alpha (T_{t-1,j} - \beta_{t-1,j} \partial |z_{(j)}^{(l)}|(1 - \alpha)^{T-t})$ and $\kappa_{T,j}^{(l)} := (1 - \alpha)^{T} (z_{0,(j)}^{(l)} - z_{(j)}^{(l)}(1 - \beta_{\gamma}^{(l)}))$. $\kappa_{T,j}^{(l)}$ decays very fast as $T$ increases. Hence, we bound the second term. We substitute $\eta_{t,j}^{(l)} = \sum_{i \neq j} \langle D_{i}^{(l)} - D_{j}^{*} \rangle (z_{i}^{*} - z_{(i)}^{*}) + \gamma_{\gamma}^{(l)}$ in $\zeta_{T,(j)}^{(l)}$.

$$
\zeta_{T,(j)}^{(l)} \in \kappa_{T,j}^{(l)} + \sum_{t=1}^{T} \alpha (T_{t-1,j} - \beta_{t-1,j} \partial |z_{(j)}^{(l)}|(1 - \alpha)^{T-t})
$$

$$
\leq \kappa_{T,j}^{(l)} + \sum_{t=1}^{T} \alpha (T_{t-1,j} - \beta_{t-1,j} \partial |z_{(j)}^{(l)}|(1 - \alpha)^{T-t}) + 2\gamma_{\gamma}^{(l)} - \beta_{t-1,j} \partial |z_{(j)}^{(l)}||(1 - \alpha)^{T-t}
$$

(63)

$$
\leq \kappa_{T,j}^{(l)} + 2\gamma_{\gamma}^{(l)} + 2(s-1) \alpha \frac{\mu \sqrt{m}}{\sqrt{t}} \sum_{t=1}^{T} E_{t-1,j} (1 - \alpha)^{T-t}
$$

27
Given above, we find an upper bound on \( E_{t-1,j}(1 - \alpha)^{T-t} \) below. From analysis in Theorem 4.4, we have

\[
E_{t-1,j} \leq v_{t-1,j} + 2(s - 1)\alpha \frac{\mu_t}{\sqrt{m}} \sum_{a=1}^{t-2} v_{a,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2} \\
+ 2(s - 1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-2}
\]

\[
E_{t-1,j}(1 - \alpha)^{T-t} \leq v_{t-1,j}(1 - \alpha)^{T-t} + 2(s - 1)\alpha \frac{\mu_t}{\sqrt{m}} \sum_{a=1}^{t-2} v_{a,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2}(1 - \alpha)^{T-t} \\
+ 2(s - 1)\alpha \frac{\mu_t}{\sqrt{m}} E_{0,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-2}(1 - \alpha)^{T-t}
\]

Re-write the first term,

\[
v_{t-1,j}(1 - \alpha)^{T-t} = (E_{0,\text{max}}(1 - \alpha)^{t-1} + (\beta_0^{(t)}|z_{(j)}^*| + 2\gamma_j^{(t)}) \sum_{k=1}^{t-1} \alpha(1 - \alpha)^{t-k-1})(1 - \alpha)^{T-t} \\
= E_{0,\text{max}}(1 - \alpha)^{T-1} + (\beta_j^{(t)}|z_{(j)}^*| + 2\gamma_j^{(t)}) \sum_{k=1}^{t-1} \alpha(1 - \alpha)^{T-k-1}
\]

Similarly,

\[
v_{a,\text{max}} = E_{0,\text{max}}(1 - \alpha)^a + (\beta_{\text{max}}^{(t)}|z_{\text{max}}^*| + 2\gamma_{\text{max}}^{(t)}) \sum_{k=1}^{a} \alpha(1 - \alpha)^{a-k}
\]

We write

\[
\sum_{a=1}^{t-2} v_{a,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2} = \sum_{a=1}^{t-2} E_{0,\text{max}}(1 - \alpha)^a(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2} \\
+ \sum_{a=1}^{t-2} (\beta_{\text{max}}^{(t)}|z_{\text{max}}^*| + 2\gamma_{\text{max}}^{(t)}) \sum_{k=1}^{a} \alpha(1 - \alpha)^{a-k}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2} \\
\leq \sum_{a=1}^{t-2} E_{0,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-2} \\
+ (\beta_{\text{max}}^{(t)}|z_{\text{max}}^*| + 2\gamma_{\text{max}}^{(t)}) \sum_{a=1}^{t-2} (1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2}
\]

Hence,

\[
(1 - \alpha)^{T-t} \sum_{a=1}^{t-2} v_{a,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2} \leq \sum_{a=1}^{t-2} E_{0,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-2}(1 - \alpha)^{T-t} \\
+ (\beta_{\text{max}}^{(t)}|z_{\text{max}}^*| + 2\gamma_{\text{max}}^{(t)}) \sum_{a=1}^{t-2} (1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{t-a-2}(1 - \alpha)^{T-t} \\
\leq (t - 2) E_{0,\text{max}}(1 - \alpha + 2\alpha \frac{\mu_t}{\sqrt{m}})^{T-2} \\
+ (\beta_{\text{max}}^{(t)}|z_{\text{max}}^*| + 2\gamma_{\text{max}}^{(t)}) \frac{(1 - \alpha)^{T-t}}{\alpha(1 - 2 \frac{\mu_t}{\sqrt{m}})}
\]
Combining all terms, we get

\[
E_{t-1,j}(1-\alpha)^{T-t} \leq v_{t-1,j}(1-\alpha)^{T-t} + 2(s-1)\alpha \frac{\mu}{\sqrt{m}} \sum_{a=1}^{t-2} v_{a,\text{max}}(1-\alpha + 2\alpha \frac{\mu}{\sqrt{m}})^{t-a-2}(1-\alpha)^{T-t}
\]

\[
+ 2(s-1)\alpha \frac{\mu}{\sqrt{m}} E_{0,\text{max}}(1-\alpha + 2\alpha \frac{\mu}{\sqrt{m}})^{t-2}(1-\alpha)^{T-t}
\]

\[
\leq E_{0,j}(1-\alpha)^{T-1} + 2(s-1)\alpha \frac{\mu}{\sqrt{m}} (t-1) E_{0,\text{max}} \delta_{\alpha,T-2}
\]

\[
+ (\beta_{\text{max}}^{(l)} |z_{\text{max}}^{*}| + 2\gamma_{\text{max}}^{(l)}) \left( \sum_{k=1}^{t-1} \alpha (1-\alpha)^{T-k-1} + 2(s-1)\kappa_l(1-\alpha)^{T-t} \right)
\]

\[
\leq (1 + 2(s-1)\alpha \frac{\mu}{\sqrt{m}}(t-1)) E_{0,\text{max}} \delta_{\alpha,T-2}
\]

\[
+ (\beta_{\text{max}}^{(l)} |z_{\text{max}}^{*}| + 2\gamma_{\text{max}}^{(l)}) \left( \sum_{k=1}^{t-1} \alpha (1-\alpha)^{T-k-1} + 2(s-1)\kappa_l(1-\alpha)^{T-t} \right)
\]

where \( \kappa_l = \frac{\mu}{(1-2\alpha \frac{\mu}{\sqrt{m}})} \). Moreover, we bound

\[
\sum_{t=1}^{T} \sum_{k=1}^{t-1} \alpha (1-\alpha)^{T-k-1} = \sum_{t=1}^{T} \alpha (1-\alpha)^{T-t} \frac{1-(1-\alpha)^{t-1}}{\alpha} \leq \sum_{t=1}^{T} (1-\alpha)^{T-t} \leq \frac{1}{\alpha}
\]

Finally, we are ready to write the bound for \( \zeta_{T,j}^{(l)} \)

\[
|\zeta_{T,j}^{(l)}| \leq \kappa_{T,j}^{(l)} + 2|\gamma_{j}^{(l)}| + 2(s-1)\alpha \frac{\mu}{\sqrt{m}} (\beta_{\text{max}}^{(l)} |z_{\text{max}}^{*}| + 2\gamma_{\text{max}}^{(l)})(1 + 2(s-1)\kappa_l)
\]

\[
+ \sum_{t=1}^{T} 2(s-1)\alpha \frac{\mu}{\sqrt{m}} ((1 + 2(s-1)\alpha \frac{\mu}{\sqrt{m}}(t-1)) E_{0,\text{max}} \delta_{\alpha,T-2})
\]

Given \( |\gamma_{j}^{(l)}| = a_{\gamma} \) with probability of \( 1 - \epsilon_{\gamma}^{(l)} \) where \( a_{\gamma} = \sqrt{\delta_{\gamma}} \) and \( s = \mathcal{O}^*(\sqrt{m}/\mu \log m) \), we will have

\[
|\zeta_{T,j}^{(l)}| \leq \mathcal{O}(\sqrt{s}||D_{j}^{*} - D_{j}^{(l)}||_{2})
\]

**Theorem 4.5** (Global forward pass code error with fixed \( \lambda_l \)). *Given Assumptions 3.3 and 3.4, suppose \( D^{(l)} \) is \( \mu \)-incoherent and \( \delta_{\gamma} = \mathcal{O}^*(1/\log p) \) close to \( D^{*} \). If \( s = \mathcal{O}^*(\sqrt{m}/\mu \log m) \), \( \mu = \mathcal{O}(\log m) \), and the regularizer and step size are chosen such that \( \lambda_l^{(l)} = \lambda^{\text{fixed}} = \frac{\mu}{\sqrt{m}} ||z^{*} - z_{0}||_{1} + a_{\gamma} = \Omega(\frac{\log m}{\sqrt{m}}) \) and \( \alpha^{(l)} \leq 1 - 2\frac{\lambda^{(l)}(1-\gamma^{2})C_{\gamma}^{\min}}{\lambda_{j,T-1}^{(l)}} \), then with probability of at least \( 1 - \epsilon_{\supp-pres}^{(l)} \) for \( j \in \supp(z^{*}) \), the code coefficient error is

\[
|z_{T,j}^{(l)} - z_{j}^{*}| \leq \mathcal{O}(\sqrt{s}||D_{j}^{(l)} - D_{j}^{*}||_{2} + \epsilon_{T,j}^{(l)\text{unroll, fixed}} + \lambda^{\text{fixed}})
\]

and

\[
z_{T,j}^{(l)} = z_{j}^{*}(1 - \beta_{j}^{(l)}) + \zeta_{T,j}^{(l)}
\]

where \( \epsilon_{T,j}^{(l)\text{unroll, fixed}} := (s-1)\alpha \frac{\mu}{\sqrt{m}} \max_{i} |z_{i}^{(l)} - z_{i}^{*}| \delta_{\alpha,T-1}^{\text{fixed}} + |z_{0,j}^{(l)} - z_{j}^{*}| \delta_{\alpha,T}^{\text{fixed}} \), \( \beta_{j}^{(l)} := (1 - \alpha + \mu \frac{\mu}{\sqrt{m}})^{T} \), \( \zeta_{T,j}^{(l)} = \mathcal{O}(a_{\gamma} + \lambda^{\text{fixed}}) \) with \( a_{\gamma} = \mathcal{O}(\sqrt{\delta_{\gamma}}) \), \( \beta_{j}^{(l)} := \langle D_{j}^{*} - D_{j}^{(l)}, D_{j}^{*} \rangle \leq \frac{\delta_{T}^{2}}{\sqrt{T}} \), and \( \epsilon_{\supp-pres}^{(l)} := \epsilon_{\supp-rec}^{(l)} + \epsilon_{\gamma}^{(l)} = 2p \exp\left(\frac{C_{\gamma}^{\min} \lambda^{\text{fixed}}}{2\gamma(\delta_{\gamma})}\right) + 2s \exp\left(\frac{1}{\delta_{\gamma}}\right) \). With appropriately large \( t \), \( |z_{T,j}^{(l)} - z_{j}^{*}| = \mathcal{O}(\sqrt{s}||D_{j}^{(l)} - D_{j}^{*}||_{2} + \lambda^{\text{fixed}}) \).
Theorem 4.2. We define

\[ E_{t,i} := |z_{t,i}^* - z_{t,i}| \]

where \( E_{t,i} \) and upper bound it as

\[
\hat{n}_{t,j}^{(l)} = \frac{\mu_t}{\sqrt{m}} \sum_{i \neq j} E_{t,i} + \gamma_j^{(l)}
\]

(73)

Opening up the recursion, we get

\[
E_{t+1,j} \leq E_{0,j} \prod_{q=0}^{t} (1 - \alpha) + \beta_j^{(l)} (z_{t+1,j}^*) \sum_{a=1}^{t+1} \alpha \prod_{q=a}^{t} (1 - \alpha) + \sum_{a=1}^{t+1} \alpha (\hat{n}_{a-1,j}^{(l)} + \lambda_a^{(l)} - \gamma_j^{(l)}) \prod_{q=a}^{t+1} (1 - \alpha)
\]

(75)

where we define \( \prod_{q=a}^{t} (1 - \alpha) = 1 \). Using the upper bounds from (73) and \( \lambda_a^{(l)} \) from Theorem 4.2, we get

\[
E_{t+1,j} \leq v_{t+1,j} + r_{t+1,j} + \sum_{a=1}^{t+1} \alpha \frac{\mu_t}{\sqrt{m}} \sum_{i \neq j} E_{a-1,i} \prod_{q=a}^{t+1} (1 - \alpha)
\]

(76)

where \( v_{t+1,j} := E_{0,j} \prod_{q=0}^{t} (1 - \alpha) + \beta_j^{(l)} (z_{t+1,j}^*) \prod_{q=0}^{t} (1 - \alpha) \) and \( r_{t+1,j} := \sum_{a=1}^{t+1} \alpha \lambda_a^{(l)} \prod_{q=a}^{t+1} (1 - \alpha) \). Following similar steps in Theorem 4.4, we now derive the general upper bound on \( E_{t+1,j} \) as follows

\[
E_{1,i_1} \leq v_{1,i_1} + r_{1,i_1} + \alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_2 \neq i_1} E_{0,i_2}
\]

(77)

For \( E_{2,i_1} \), we have

\[
E_{2,i_1} \leq v_{2,i_1} + r_{2,i_1} + \alpha \frac{\mu_t}{\sqrt{m}} \left( \sum_{i_2 \neq i_1} E_{1,i_2} + \sum_{i_2 \neq i_1} E_{0,i_2} (1 - \alpha) \right)
\]

(78)

Substituting \( E_{1,i_1} \),

\[
E_{2,i_1} \leq v_{2,i_1} + r_{2,i_1} + \alpha \frac{\mu_t}{\sqrt{m}} \left( \sum_{i_2 \neq i_1} (v_{1,i_2} + r_{1,i_2} + \alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_3 \neq i_2} E_{0,i_3}) + \sum_{i_2 \neq i_1} E_{0,i_2} (1 - \alpha) \right)
\]

(79)

For \( E_{3,i_1} \), we have

\[
E_{3,i_1} \leq v_{3,i_1} + r_{3,i_1} + \alpha \frac{\mu_t}{\sqrt{m}} \sum_{a=1}^{3} \sum_{i_2 \neq i_1} E_{a-1,i_2} (1 - \alpha)^{3-a}
\]

(80)

Unrolling the recursion,

\[
E_{3,i_1} \leq v_{3,i_1} + r_{3,i_1} + \alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_2 \neq i_1} (v_{2,i_2} + r_{2,i_2}) + \alpha \frac{\mu_t}{\sqrt{m}} \left( (1 - \alpha) \sum_{i_2 \neq i_1} (v_{1,i_2} + r_{1,i_2}) + \alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_3 \neq i_2} (v_{1,i_3} + r_{1,i_3}) \right)
\]

\[
+ \alpha \frac{\mu_t}{\sqrt{m}} \left( (1 - \alpha)^2 \sum_{i_2 \neq i_1} E_{0,i_2} + 2(1 - \alpha) (\alpha \frac{\mu_t}{\sqrt{m}} \sum_{i_3 \neq i_2,i_1} E_{0,i_3}) + (\alpha \frac{\mu_t}{\sqrt{m}})^2 \sum_{i_3 \neq i_2,i_1} E_{0,i_4} \right)
\]

(81)
Given above, we can write up the relation as

\[
E_{3,i} \leq v_{3,i} + r_{3,i} + \alpha \frac{\mu_i}{\sqrt{m}} \left( (s-1)(v_{2,i} + r_{2,i}) + (v_{1,i} + r_{1,i})(1-\alpha)(s-1) + \alpha \frac{\mu_i}{\sqrt{m}} (s-2) \right) \\
+ \alpha \frac{\mu_i}{\sqrt{m}} E_{0,i} \left( (s-1)(1-\alpha)^2 + 2(1-\alpha)(\alpha \frac{\mu_i}{\sqrt{m}})(s-2) + (\alpha \frac{\mu_i}{\sqrt{m}})^2 (s-3) \right) \\
\leq v_{3,i} + r_{3,i} + (s-1)\alpha \frac{\mu_i}{\sqrt{m}} \left( v_{2,i} + r_{2,i} + (v_{1,i} + r_{1,i})(1-\alpha) + \alpha \frac{\mu_i}{\sqrt{m}} \right) \\
+ (s-1)\alpha \frac{\mu_i}{\sqrt{m}} E_{0,i} \left( (1-\alpha)^2 + 2(1-\alpha)(\alpha \frac{\mu_i}{\sqrt{m}}) + (\alpha \frac{\mu_i}{\sqrt{m}})^2 \right)
\]

(82)

We denote \( u_{t+1,i} := v_{t+1,i} + r_{t+1,i} \) and following similar steps for \( E_{4,i} \), we get,

\[
E_{4,i} \leq u_{4,i} + (s-1)\alpha \frac{\mu_i}{\sqrt{m}} \left( u_{3,i} + u_{2,i}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}}) + u_{1,i}(1-\alpha)^2 + 2(1-\alpha)(\alpha \frac{\mu_i}{\sqrt{m}}) + (\alpha \frac{\mu_i}{\sqrt{m}})^2 \right) \\
+ (s-1)\alpha \frac{\mu_i}{\sqrt{m}} E_{0,i} \left( (1-\alpha)^3 + 3(1-\alpha)(\alpha \frac{\mu_i}{\sqrt{m}})^2 + 3(1-\alpha)^2(\alpha \frac{\mu_i}{\sqrt{m}})^3 \right)
\]

(83)

This leads to the term

\[
E_{4,i} \leq u_{4,i} + (s-1)\alpha \frac{\mu_i}{\sqrt{m}} \left( u_{3,i} + u_{2,i}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^1 + u_{1,i}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^2 \right) \\
+ (s-1)\alpha \frac{\mu_i}{\sqrt{m}} E_{0,i}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^3
\]

(84)

Hence, the general term for code error at \( t \) layer is

\[
E_{t+1,j} \leq u_{t+1,j} + (s-1)\alpha \frac{\mu_i}{\sqrt{m}} \sum_{a=1}^{t} (v_{a,\text{max}} + r_{a,\text{max}})(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} + (s-1)\alpha \frac{\mu_i}{\sqrt{m}} E_{0,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^t
\]

(85)

where for \( j \) in the support, we define the upper bounds \( v_{a,j} \leq v_{a,\text{max}} \), \( r_{a,j} \leq r_{a,\text{max}} \), and \( E_{0,j} \leq E_{0,\text{max}} \). Next, we define \((1-\alpha)^t \leq \delta_{\text{fixed}} = (1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^t\), and use it to find an upper bound on the two expressions \( \sum_{a=1}^{t} v_{a,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \) and \( \sum_{a=1}^{t} r_{a,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \). The following bound can be achieved similar to the steps in Theorem 4.4

\[
\sum_{a=1}^{t} r_{a,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \leq E_{0,\text{max}} \delta^{\text{fixed}}_{a,t} + \frac{1}{\alpha(1-\mu_i/\sqrt{m})} \left( \frac{\delta^2}{2} |z_{\text{max}}^*| + a_\gamma \right)
\]

(86)

Hence, we focus on \( \sum_{a=1}^{t} r_{a,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \) next. First, we rewrite

\[
r_{t+1,j} = \sum_{a=1}^{t+1} \alpha \lambda_{a-1,j}^{\text{fixed}} \prod_{q=a}^{t+1} (1-\alpha)
\]

(87)

We replace all \( \lambda_{a,j}^{\text{fixed}} \) with a fixed one \( \lambda^{\text{fixed}} \), and write

\[
\sum_{a=1}^{t} r_{a,j}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \leq \sum_{a=1}^{t} \lambda^{\text{fixed}} \sum_{k=1}^{a} \alpha(1-\alpha)^{a-k+1}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \\
\leq \lambda^{\text{fixed}} \sum_{a=1}^{t} (1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \sum_{k=1}^{a} \alpha(1-\alpha)^{a-k+1}
\]

(88)

Using sum of geometric series, we write \( \sum_{a=1}^{t}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} = \frac{1-(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^t}{\alpha - \alpha \frac{\mu_i}{\sqrt{m}}} \leq \frac{1}{\alpha(1-\mu_i/\sqrt{m})} \). Hence, we get

\[
\sum_{a=1}^{t} r_{a,\text{max}}(1-\alpha + \alpha \frac{\mu_i}{\sqrt{m}})^{t-a} \leq \lambda^{\text{fixed}} \frac{1}{\alpha(1-\mu_i/\sqrt{m})}
\]

(89)
Hence, we bound the code error on the coefficients as following

\[ E_{t+1,j} \leq u_{t+1,j} + (s-1) \frac{\mu l}{\sqrt{m}} \left( \frac{1}{1 - \frac{n}{\sqrt{m}}} \right) \alpha \frac{\delta^2}{2} |z^*_t| + a_\gamma + \lambda^{\text{fixed}} \right) + (t+1)(s-1)\alpha \frac{\mu l}{\sqrt{m}} E_{0,\text{max},\theta_{t+1}^{\text{fixed}}} \tag{90} \]

Next, we further simplify the first term. From before, we have

\[ \nu_{t+1,j} = E_{0,j}(1-\alpha)^{t+1} + (\beta_j^{(l)}|z^*_j| + \gamma_j^{(l)}) \sum_{k=1}^{t+1} \alpha(1-\alpha)^{t-k+1} \leq E_{0,j}\theta_{t+1}^{\text{fixed}} + \frac{\delta^2}{2} |z^*_\text{max}| + a_\gamma \tag{91} \]

and

\[ r_{t+1,j} = \alpha \lambda^{\text{fixed}} \sum_{k=1}^{t+1} (1-\alpha)^{t-k+1} \leq \lambda^{\text{fixed}} \tag{92} \]

Substituting the above upper bound into the upper bound for \( E_{t+1,j} \), we get

\[ E_{t+1,j} \leq E_{0,j}\theta_{t+1}^{\text{fixed}} + (1 + (s-1)\kappa_j^{\text{fixed}})(\frac{\delta^2}{2} |z^*_\text{max}| + a_\gamma + \lambda^{\text{fixed}}) + (t+1)(s-1)\alpha \frac{\mu l}{\sqrt{m}} E_{0,\text{max},\theta_{t+1}^{\text{fixed}}} \tag{93} \]

where \( \kappa_j^{\text{fixed}} := \frac{\mu l}{\sqrt{m}} \left( \frac{1}{1 - \frac{n}{\sqrt{m}}} \right) \). Given \( s = \mathcal{O}^*(\sqrt{m}/\mu \log m) \), we have \( (s-1)\kappa_j^{\text{fixed}} < 1 \). Hence, with probability of at least \( 1 - \epsilon_j^{(l)} \), we have

\[ E_{t,j} \leq \mathcal{O}(a_\gamma + \lambda^{\text{fixed}}) + (s-1)t\alpha \frac{\mu l}{\sqrt{m}} E_{0,\text{max},\theta_{t-1}^{\text{fixed}}} + E_{0,j}\theta_{t+1}^{\text{fixed}} \tag{94} \]

where \( \epsilon_j^{(l),\text{unroll},\text{fixed}} := (s-1)\alpha t \frac{\mu l}{\sqrt{m}} \max_i |z_j^{(l)} - z^*_j|^{\text{fixed}} + |z_j^{(l)} - z^*_j|^{\text{fixed}} \). With appropriately large unrolled layer \( t \), \( \epsilon_j^{(l),\text{unroll},\text{fixed}} \approx 0 \). Hence, for the code error on non-zero coefficients, we get

\[ |z_j^{(l)} - z^*_j| = \mathcal{O}(\sqrt{s\|D_j - D_j\|_2} + \lambda^{\text{fixed}}) \tag{95} \]

for large enough \( t \). Now, we provide the relation for \( z_{T,j} \). We re-write (35)

\[ z_{T,j} \in z^*_j(1 - \beta_j^{(l)}) + \xi_{T,j}^{(l)} \tag{96} \]

where \( \xi_{T,j}^{(l)} = \kappa_{T,j}^{(l)} + \sum_{t=1}^{T} \alpha(\eta_{t-1,j} - \lambda_{t-1,j} \partial z_{t,j}(\alpha)) (1 - \alpha)^{-t} \) and \( \kappa_{T,j}^{(l)} := (1 - \alpha)^T (z_{0,j} - z^*_j)(1 - \beta_j^{(l)}) \). \( \kappa_{T,j}^{(l)} \) decays very fast as \( T \) increases. Hence, we bound the second term. We substitute \( \eta_{t,j}^{(l)} = \sum_{i \neq j} \langle D_i^{(l)}, D_j^{(l)} \rangle (z^*_i - z_{t-1,i}) \) and \( \gamma_j^{(l)} = \lambda^{\text{fixed}} \).

\[ \xi_{T,j}^{(l)} \leq \kappa_{T,j}^{(l)} + \gamma_j^{(l)} + \lambda^{\text{fixed}} \sum_{t=1}^{T} (s-1)(1-\alpha)^{-t} + \alpha \frac{\mu l}{\sqrt{m}} \sum_{t=1}^{T} E_{t-1,j}(1 - \alpha)^{-t} \tag{97} \]

\[ \leq \kappa_{T,j}^{(l)} + \gamma_j^{(l)} + \lambda^{\text{fixed}} + (s-1)(1-\alpha)^{-t} + \alpha \frac{\mu l}{\sqrt{m}} \sum_{t=1}^{T} E_{t-1,j}(1 - \alpha)^{-t} \]
Given above, we find an upper bound on $E_{t-1,j}(1-\alpha)^{T-t}$ below. From analysis in Theorem 4.4, we have

$$E_{t-1,j} \leq v_{t-1,j} + (s-1)\alpha \frac{\mu_l}{\sqrt{m}} \sum_{a=1}^{t-2} v_{a,\max}(1 - \alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}$$

$$+ (s-1)\alpha \frac{\mu_l}{\sqrt{m}} E_{0,\max}(1 - \alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-2}$$

$$E_{t-1,j}(1-\alpha)^{T-t} \leq E_{t-1,j}(1-\alpha)^{T-t} + 2(s-1)\alpha \frac{\mu_l}{\sqrt{m}} \sum_{a=1}^{t-2} v_{a,\max}(1 - \alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}(1-\alpha)^{T-t}$$

$$+ (s-1)\alpha \frac{\mu_l}{\sqrt{m}} E_{0,\max}(1 - \alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-2}(1-\alpha)^{T-t}$$

Re-write the first term,

$$v_{t-1,j}(1-\alpha)^{T-t} = (E_{0,j}(1-\alpha)^{t-1} + (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{k=1}^{t-1} \alpha(1-\alpha)^{t-k-1})(1-\alpha)^{T-t}$$

$$= E_{0,j}(1-\alpha)^{T-1} + (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1}$$

Similarly,

$$v_{a,\max} = E_{0,\max}(1 - \alpha)^a + (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{k=1}^{a} \alpha(1-\alpha)^a-k$$

We write

$$\sum_{a=1}^{t-2} v_{a,\max}(1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2} = \sum_{a=1}^{t-2} E_{0,\max}(1 - \alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}$$

$$+ \sum_{a=1}^{t-2} (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{k=1}^{a} \alpha(1-\alpha)^a-k(1-\alpha - \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}$$

$$\leq \sum_{a=1}^{t-2} E_{0,\max}(1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-2}$$

$$+ (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{a=1}^{t-2} (1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}$$

Hence,

$$(1-\alpha)^{T-t} \sum_{a=1}^{t-2} v_{a,\max}(1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2} \leq \sum_{a=1}^{t-2} E_{0,\max}(1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-2}(1-\alpha)^{T-t}$$

$$+ (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \sum_{a=1}^{t-2} (1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{t-a-2}(1-\alpha)^{T-t}$$

$$\leq (t-2)E_{0,\max}(1 - \alpha + \alpha \frac{\mu_l}{\sqrt{m}})^{T-2}$$

$$+ (\beta_j^{(l)}|z_j^*| + \gamma_j^{(l)} + \lambda^{\text{fixed}}) \frac{(1-\alpha)^{T-t}}{\alpha(1-\frac{\mu_l}{\sqrt{m}})}$$

(102)
Combining all terms, we get

\[
E_{t-1,j}(1-\alpha)^{T-t} \leq v_{t-1,j}(1-\alpha)^{T-t} + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}\sum_{a=1}^{t-2}v_{a,\text{max}}(1-\alpha + \alpha\frac{\mu_l}{\sqrt{m}})^{t-a-2}(1-\alpha)^{T-t} + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}E_{0,\text{max}}(1-\alpha + \alpha\frac{\mu_l}{\sqrt{m}})^{t-2}(1-\alpha)^{T-t}
\]

\[
\leq E_{0,j}(1-\alpha)^{T-1} + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}(t-1)E_{0,\text{max}}\delta_{a,T-2} + (\beta^{(l)}_{\text{max}}|z_{\text{max}}^*| + \gamma^{(l)}_{\text{max}} + \lambda^{\text{fixed}}) \left(\sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1} + (s-1){\kappa}_i(1-\alpha)^{T-t}\right)
\]

\[
\leq (1 + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}(t-1))E_{0,\text{max}}\delta_{a,T-2} + (\beta^{(l)}_{\text{max}}|z_{\text{max}}^*| + \gamma^{(l)}_{\text{max}} + \lambda^{\text{fixed}}) \left(\sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1} + (s-1){\kappa}_i(1-\alpha)^{T-t}\right)
\]

where \(\kappa_i^{\text{fixed}} = \frac{\mu_l}{(t-2)\sqrt{m}}\). Moreover, we bound

\[
\sum_{t=1}^{T} \sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1} = \sum_{t=1}^{T} \alpha(1-\alpha)^{T-t}1 - (1-\alpha)^{T-1} \frac{1}{\alpha} \leq \sum_{t=1}^{T} (1-\alpha)^{T-t} \leq \frac{1}{\alpha}
\]

Finally, we are ready to write the bound for \(\zeta^{(l)}_{T,j}\)

\[
|\zeta^{(l)}_{T,j}| \leq \kappa^{(l)}_{T,j} + |\gamma^{(l)}_j| + \lambda^{\text{fixed}} + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}(\beta^{(l)}_{\text{max}}|z_{\text{max}}^*| + \gamma^{(l)}_{\text{max}} + \lambda^{\text{fixed}})(1 + (s-1){\kappa}_i^{\text{fixed}})
\]

\[
+ \sum_{t=1}^{T} (s-1)\alpha\frac{\mu_l}{\sqrt{m}}((1 + (s-1)\alpha\frac{\mu_l}{\sqrt{m}}(t-1))E_{0,\text{max}}\delta_{a,T-2} + (\beta^{(l)}_{\text{max}}|z_{\text{max}}^*| + \gamma^{(l)}_{\text{max}} + \lambda^{\text{fixed}}) \left(\sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1} + (s-1){\kappa}_i(1-\alpha)^{T-t}\right)
\]

Given \(|\gamma^{(l)}_j| = a_\gamma\) with probability of \(1 - \epsilon_\gamma\) where \(a_\gamma = \sqrt{3d_\gamma}\) and \(s = O(\sqrt{m/\mu \log m})\), we will have

\[
|\zeta^{(l)}_{T,j}| \leq O(\sqrt{s\|D^{(l)}_j\| - D_j^*\|_2} + \lambda^{\text{fixed}})
\]

We now re-state the forward pass Jacobian (Theorem 4.7) convergence.

**Theorem 4.7** (Local forward pass Jacobian convergence). Given the recursion \(z_{t+1} = \Phi(z_t, D)\), and \(\hat{z}\) the unique minimizer of lasso with Jacobian \(\hat{J}\), then \(\exists \rho < 1, B > 0 \text{ s.t. } \|J_t - \hat{J}\|_2 \leq O(t\rho^t) \forall t > B\). Furthermore, given Theorem 4.1 and Theorem 4.2, \(B = 1\).

**Proof.** Differentiating the recursion,

\[
J_{t+1} = \nabla_1\Phi(z_t, D)^T J_t + \nabla_2\Phi(z_t, D)^T.
\]

Similarly,

\[
\hat{J} = \nabla_1\Phi(\hat{z}, D)^T \hat{J} + \nabla_2\Phi(\hat{z}, D)^T
\]

where \(\hat{z}\) is a minimizer of lasso and fixed-point of the mapping (see Lemma 3.2). Subtract the terms

\[
J_{t+1} - \hat{J} = \nabla_1\Phi(z_t, D)^T (J_t - \hat{J}) + (\nabla_1\Phi(z_t, D) - \nabla_1\Phi(\hat{z}, D))^T \hat{J} + (\nabla_2\Phi(z_t, D) - \nabla_2\Phi(\hat{z}, D))^T \hat{J}
\]

where
Given the Lipschitz properties of \( L \) and \( h \), we can further get the upper bounds on \( \| \nabla_1 \Phi(a, D) - \nabla_1 \Phi(b, D) \|_2 \leq L_q_1 \| b - a \|_2 \) and \( \| \nabla_2 \Phi(a, D) - \nabla_2 \Phi(b, D) \|_2 \leq L_q_2 \| b - a \|_2 \). Hence, with upper bound on the norm of Jacobian (Assumption 4.1), there exists \( B > 0 \) such that \( \forall t > B \)

\[
\| J_{t+1} - \hat{J} \|_2 \leq \| \nabla_1 \Phi(z_t, D) \|_2 \| J_t - \hat{J} \|_2 + \| \nabla_1 \Phi(z_t, D) - \nabla_1 \Phi(\hat{z}, D) \|_2 \| \hat{J} \|_2 \\
+ \| \nabla_2 \Phi(z_t, D) - \nabla_2 \Phi(\hat{z}, D) \|_2 \| \hat{J} \|_2 \\
\leq c \| J_t - \hat{J} \|_2 + c \| z_t - \hat{z} \|_2
\]

where \( c \triangleq M_3 L_{q_1} + L_{q_2} \). Hence,

\[
\| J_{t+1} - \hat{J} \|_2 \leq \rho \| J_t - \hat{J} \|_2 + O(\rho^t).
\]

Unrolling the recursion,

\[
\| J_{t+1} - \hat{J} \|_2 \leq O((t+1)\rho^t).
\]

**Theorem 4.6** (Global forward pass code error). Let \( \hat{z} \) be the fixed-point of the encoder with iterations \( z_{t+1} = \Phi(z_t, D) \). Given Assumption 3.6, Lemmas 3.2, A.1 and A.2, we have \( \| \hat{z} - z^* \|_2 \leq O(\| D - D^* \|_2 + \hat{\delta}^*) \), where \( \hat{\delta}^* = \| \hat{z} - z^* \|_2 \), \( \hat{z} \) is the unique minimizer of lasso (1) given the dictionary \( D \), \( z^* \) is the unique minimizer of lasso (1) given the dictionary \( D^* \), and \( z^* \) is the ground-truth code.

**Proof.** We first find the error between \( \hat{z} \) and \( \hat{z}^* \) which is the unique minimizer of lasso (1) given the true dictionary \( D^* \). Using fixed-point property (Lemma 3.2), we get

\[
\| \hat{z} - \hat{z}^* \|_2 = \| \Phi(\hat{z}, D) - \Phi(\hat{z}^*, D^*) \|_2 \leq \| \Phi(\hat{z}, D) - \Phi(\hat{z}^*, D) \|_2 + \| \Phi(\hat{z}^*, D) - \Phi(\hat{z}^*, D^*) \|_2
\]

Using the \( \mu \)-strongly convexity of \( \mathcal{L}_x(z_t, D) \) on the support, and \( L_{21} \) Lipschitz constants of \( \nabla^2_{21} \mathcal{L}_x(z, D) \), we upper bound the term as follows:

\[
\| \hat{z} - \hat{z}^* \|_2 \leq \rho \| \hat{z} - \hat{z}^* \|_2 + c_{\text{prox}} L_{21} \| D - D^* \|_2
\]

Where \( \rho \triangleq c_{\text{prox}} (1 - \alpha \mu) < 1 \). Denote \( q \triangleq \frac{\alpha \mu c_{\text{prox}} L_{21}}{1 - \rho} \) which can be made to be small with proper choice of step size \( \alpha \).

\[
\| \hat{z} - \hat{z}^* \|_2 \leq q \| D - D^* \|_2
\]

Hence, we get the following code error

\[
\| \hat{z} - \hat{z}^* \|_2 \leq \| \hat{z} - \hat{z}^* \|_2 + \| \hat{z}^* - z^* \|_2 \leq q \| D - D^* \|_2 + \hat{\delta}^* \leq O(\| D - D^* \|_2 + \hat{\delta}^*)
\]

**Theorem 4.8** (Global forward pass Jacobian error). Let \( \hat{z} \) be the fixed-point of the encoder with iterations \( z_{t+1} = \Phi(z_t, D) \). Given Assumption 3.6, Lemmas 3.2, A.1 and A.2, we have \( \| \hat{J} - J^* \|_2 \leq O(\| D - D^* \|_2 + \hat{\delta}^*) \), where \( \hat{\delta}^* := \| \hat{J}^* - J^* \|_2 \), and \( \hat{J}, J^* \) are Jacobians corresponding to \( \hat{z}, \hat{z}^* \) and \( z^* \).

**Proof.** First, we define \( J^* \). For \( z^* \), we define the mapping function \( z \rightarrow \mathcal{L}_x(z, D) \), where \( z^*(D) \) is its minimizer evaluated at \( D^* \), i.e., \( \nabla_1 \mathcal{L}_x(z^*, D^*) = D^{*\top} (D^* z^* - x) = 0 \) given the generative model \( (x = D^* z^*) \). Hence, we define the Jacobian \( J^* = \frac{\partial z^*}{\partial D} |_{D=D^*} \). From implicit function theorem, we get

\[
J^{**} \nabla^2_{11} \mathcal{L}_x(z^*, D^*) + \nabla^2_{21} \mathcal{L}_x(z^*, D^*) = 0
\]

which is later used in the global backward pass analysis. Alternatively, if \( \nabla^2_{11} \mathcal{L}_x(z^*, D^*) \) is invertible, then we can compute \( J^{**} \) as follows:

\[
J^{**} = -\nabla^2_{21} \mathcal{L}_x(z^*, D^*) \nabla^2_{11} \mathcal{L}_x(z^*, D^*)^{-1}
\]
The Jacobian w.r.t row $i$ of the dictionary is
\[ J_{i,1}^* = -\left(D_{S^*}^T D_{S^*}^*\right)^{-1}(D_{S^*}^T z^* + (D_{S_i}^T z^* - x_i)i)_S \]
on the support $S^*$ of $z^*$. Outside of the support, it is zero. Now, given the recursion $z_{t+1} = \Phi(z_t, D)$, we differentiate the recursion,
\[ J_{t+1} = \nabla_1 \Phi(z_t, D)^T J_t + \nabla_2 \Phi(z_t, D)^T. \]
Hence, we have
\[ \hat{J} = \nabla_1 \Phi(\hat{z}, D)^T \hat{J} + \nabla_2 \Phi(\hat{z}, D)^T \]
where $\hat{J}$ is the Jacobian of $\hat{z}^*$. Then, following similar step to Theorem 4.7, we can write
\[ \hat{J} - \hat{J}^* = \nabla_1 \Phi(\hat{z}, D)^T (\hat{J} - \hat{J}^*) + (\nabla_1 \Phi(\hat{z}, D) - \nabla_1 \Phi(\hat{z}^*, D^*))^T \hat{J}^* + (\nabla_2 \Phi(\hat{z}, D) - \nabla_2 \Phi(\hat{z}^*, D^*))^T \]
With respect to $D$, we denote the Lipschitz constants of $\nabla_1 \Phi(\hat{z}, D)$ and $\nabla_2 \Phi(\hat{z}, D)$ with $L_{\Phi,1,D}$ and $L_{\Phi,2,D}$, respectively. Then,
\[ \|\hat{J} - \hat{J}^*\|_2 \leq \rho \|\hat{J} - \hat{J}^*\|_2 + c\|\hat{z} - \hat{z}^*\|_2 + c_D \|\hat{D} - D^*\|_2 \]
where $c \triangleq M_J L_{\Phi,1} + L_{\Phi,2}$ and $c_D \triangleq M_J L_{\Phi,1} + L_{\Phi,2}$. Given the global forward pass code error, we get
\[ \|\hat{J} - \hat{J}^*\|_2 \leq \rho \|\hat{z} - \hat{z}^*\|_2 + c\|D - D^*\|_2 \leq (q_D + q_z)\|D - D^*\|_2 \leq O(\|D - D^*\|_2 + \delta_J^*) \]
where we denote $\delta_J^* := \|\hat{J}^* - J^*\|_2$

### A.4 Local backward pass proof details

In each update of the dictionary, we bound the gradient approximations as function of unrolling $t$ (Theorem 4.9). This shows that $g_{t,\text{dec}}^{\text{ae-lasso}}$ converges faster than $g_{t,\text{dec}}$ and $g_{t,\text{ae-lasso}}^{\text{dec}}$, and the latter is a biased estimator of $\hat{g}$. This is followed by Theorem 4.9 showing the order magnitude of the bounds is indeed tight.

**Theorem 4.9** (Local convergence of gradients). *Given the forward pass convergence results (Theorems 4.3 and 4.7), $\exists \rho < 1, B > 0$ such that \forall $t > B$, the errors of gradients defined in Algorithm 2 w.r.t $\hat{g}$ (4) satisfy
\[ \|g_{t,\text{dec}}^{\text{ae-lasso}} - \hat{g}\|_2 \leq O(\rho^t) \]
\[ \|g_{t,\text{dec}}^{\text{ae-lasso}} - \hat{g}\|_2 \leq O(\rho^t) \]
\[ \|g_{t,\text{ae-lasso}}^{\text{dec}} - \hat{g}\|_2 \leq O(t \rho^t + M_J \sqrt{s}). \]

Moreover, the order of upper bounds is tight (see Lemma A.4).

**Proof.** For $g_{t,\text{dec}}$ with the infinite fresh samples, we have $\lim_{n \to \infty} \frac{1}{\sqrt{n}} \sum_{i=1}^n \nabla_2 L_{\mathcal{X}}(z_i, D) = E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(z_i, D) \right]$ a.s. Based on Lemma 3.3, we get
\[ \|g_{t,\text{dec}} - \hat{g}\|_2 = \|E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(z_i, D) \right] - E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(\hat{z}, D) \right]\|_2 \leq E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(z_i, D) - \nabla_2 L_{\mathcal{X}}(\hat{z}, D) \right]_2 \leq E_{x \in \mathcal{X}} \left[ L_2 \|z_i - \hat{z}\|_2 \right] \leq O(\rho^t). \]

Similarly, for $g_{t,\text{ae-lasso}}$ and $g_{t,\text{ae-lasso}}$, we replace the sample mean for gradient computations with expectation in their limit. We re-write the gradient estimation error as following
\[ \hat{g} = E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(\hat{z}, J_i)(z_i - \hat{z}) \right] + E_{x \in \mathcal{X}} \left[ \nabla_2 L_{\mathcal{X}}(\hat{z}, D) - \nabla_2 L_{\mathcal{X}}(\hat{z}, D) \right] \leq E_{x \in \mathcal{X}} \left[ L_2 \|z_i - \hat{z}\|_2 \right] \leq O(\rho^t). \]

\[ \hat{g} = E_{x \in \mathcal{X}} \left[ Q(\hat{z}, J_i)(z_i - \hat{z}) \right] + E_{x \in \mathcal{X}} \left[ Q^{11}(\hat{z}) \right] + E_{x \in \mathcal{X}} \left[ J_i Q^{\text{lasso-11}}(\hat{z}) \right] \]

\[ \hat{g} = E_{x \in \mathcal{X}} \left[ Q(\hat{z}, J_i)(z_i - \hat{z}) \right] + E_{x \in \mathcal{X}} \left[ Q^{11}(\hat{z}) \right] + E_{x \in \mathcal{X}} \left[ J_i Q^{\text{ae-11}}(\hat{z}) \right] \]
where
\[
Q_t^{21}(z) \equiv \nabla_2 \mathcal{L}_x(z, D) - \nabla_2 \mathcal{L}_x(z, D) - \nabla_2^2 \mathcal{L}_x(z, D)(z_t - z)
\]
\[
Q_t^{\text{lasso-11}}(z) \equiv \nabla_1 \mathcal{L}_x(z, D) + \partial h(z_t) - \nabla_1^2 \mathcal{L}_x(z, D)(z_t - z)
\]
\[
Q_t^{\text{lasso-11}}(z) \equiv \nabla_1 \mathcal{L}_x(z, D) - \nabla_1^2 \mathcal{L}_x(z, D)(z_t - z)
\]
\[
Q(z, J) \equiv J^+ \nabla_1^2 \mathcal{L}_x(z, D) + \nabla_2^2 \mathcal{L}_x(z, D).
\]

We provide bounds on the above in Lemma A.5. Hence, it suffices to bound the terms on the r.h.s as follows:
\[
\|g_t^{\text{ae-lasso}} - \hat{g}\|_2 \leq E_{x \in \mathcal{X}}[L_1 \|J_t - \hat{J}\|_2 \|z_t - \hat{z}\|_2 + (L_{21}/2)\|z_t - \hat{z}\|^2 + M_J((L_{11}/2)\|z_t - \hat{z}\|^2 + \|\partial h(\hat{z})\|_2)].
\]

Using the convergence errors from the forward pass (Theorems 4.3 and 4.7),
\[
\|g_t^{\text{ae-lasso}} - \hat{g}\|_2 \leq L_1 O(t \rho^{2t}) + (L_{21}/2 + M_J(L_{11}/2)) O(\rho^{2t}) = O(t \rho^{2t}).
\]

Similarly,
\[
\|g_t^{\text{ae-ls}} - \hat{g}\|_2 \leq L_1 O(t \rho^{2t}) + (L_{21}/2 + M_JL_{11}/2) O(\rho^{2t}) + M_J\|\partial h(\hat{z})\|_2 = O(t \rho^{2t} + M_J \lambda \sqrt{z}).
\]

Lemma A.4 (Tight local bound). The order magnitude of the upper bounds in Theorem 4.9 is tight.

Proof. It is sufficient to show that there exist an example such that its forward pass code and Jacobian convergences are $O(\rho^t)$ and $O(t \rho^t)$, respectively. The following example confirms this. Without loss of generality, let $z^*$ be 1-sparse and non-negative, $D = D^*$ and $D_j = 0$ for $j \neq i$. The loss function is
\[
\frac{1}{2} \|D_i z_t(i) - D_i z^*(i)\|^2 + \lambda \|z_t\|.
\]

Hence, the forward pass convergences are
\[
z_{t+1, (i)} = z_{t, (i)} - \alpha (D_i^T D_i z_{t, (i)} + D_i^T z^*_{(i)}) + \lambda = (1 - \alpha) z_{t, (i)} + \alpha (z^*_{(i)} - \lambda).
\]

Hence, the forward pass convergences are
\[
z_{t, (i)} = (1 - \alpha)^t z_0 + \sum_{k=1}^t (1 - \alpha)^{-k} (z^*_{(i)} - \lambda) = (1 - \alpha)^t z_0 + (1 - (1 - \alpha)^t) (z^*_{(i)} - \lambda)
\]
\[
z_{t, (i)} - \hat{z}_{(i)} = \rho^t (z_0 - z^*_{(i)} + \lambda) = O(\rho^t)
\]
\[
J_{t, (i)} = J_{t-1, (i)} - \alpha (J_{t-1, (i)} + 2D_i z_{t, (i)} - D_i^* z_{*_{(i)}}) = \rho J_{t-1, (i)} + O(\rho^t) + \tilde{J}_{(i)}
\]
\[
J_{t, (i)} - \hat{J}_{(i)} = \rho^t \tilde{J}_{0, (i)} + \sum_{k=1}^t O(\rho^t) = O(t \rho^t)
\]

where $\rho = 1 - \alpha$, $\hat{z}_{(i)} = z^*_{(i)} - \lambda$, and $\tilde{J}_{(i)} = \alpha (2D_i \hat{z}_{(i)} - D_i^* z^*_{(i)})$.

Lemma A.5 (Local bounds). From local gradient errors in Theorem 4.9, the following are satisfied
\[
\|Q_t^{21}(\hat{z})\|_2 \leq (L_{21}/2)\|z_t - \hat{z}\|^2, \quad \|Q_t^{\text{lasso-11}}(\hat{z})\|_2 \leq (L_{11}/2)\|z_t - \hat{z}\|^2
\]
\[
\|Q(\hat{z}, J_t)\|_2 \leq L_1 \|J_t - \hat{J}\|_2, \quad \|Q_t^{\text{ae-lasso-11}}(\hat{z})\|_2 \leq (L_{11}/2)\|z_t - \hat{z}\|^2 + \|\partial h(\hat{z})\|_2.
\]

Proof. For $Q_t^{21}(\hat{z})$, given convexity of $\nabla_1 \mathcal{L}_x(z, D)$ and its domain (Assumption 3.1) and Lemma 3.3, we achieve the quadratic upper bound. For $Q_t^{\text{ae-lasso-11}}(\hat{z})$, we add and subtract $\nabla_1 \mathcal{L}_x(\hat{z}, D)$, and then use quadratic
upper bound. At line four, given Lemma A.3, we use \(0 \in \nabla_1 \mathcal{L}_x(\hat{z}, D) + \partial h(\hat{z})\) and assume that \(z_t\) recovers the sign entries of \(\hat{z}\).

\[
\|Q_1^{\text{lasso}}\|_2 = \|\nabla_1 \mathcal{L}_x(z_t, D) + \partial h(z_t) - \nabla^2_1 \mathcal{L}_x(\hat{z}, D)(z_t - \hat{z})\|_2 \\
= \|\nabla_1 \mathcal{L}_x(z_t, D) - \nabla_1 \mathcal{L}_x(\hat{z}, D) + \nabla_1 \mathcal{L}_x(\hat{z}, D) + \partial h(z_t) - \nabla^2_1 \mathcal{L}_x(\hat{z}, D)(z_t - \hat{z})\|_2 \\
\leq (\|z_t - \hat{z}\|_2^2 + \|\partial h(z_t) + \nabla_1 \mathcal{L}_x(\hat{z}, D)\|_2) \\
\leq (\|z_t - \hat{z}\|_2^2 + \|\partial h(\hat{z})\|_2) \tag{123}
\]

Similarly,

\[
\|Q_1^{\text{ls}}\|_2 = \|\nabla_1 \mathcal{L}_x(z_t, D) - \nabla^2_1 \mathcal{L}_x(\hat{z}, D)(z_t - \hat{z})\|_2 \\
= \|\nabla_1 \mathcal{L}_x(z_t, D) - \nabla_1 \mathcal{L}_x(\hat{z}, D) + \nabla_1 \mathcal{L}_x(\hat{z}, D) - \nabla^2_1 \mathcal{L}_x(\hat{z}, D)(z_t - \hat{z})\|_2 \\
\leq (\|z_t - \hat{z}\|_2^2 + \|\nabla_1 \mathcal{L}_x(\hat{z}, D)\|_2) \tag{124}
\]

For \(Q(\hat{z}, J)\), from implicit function theorem, \(Q(\hat{z}, J) = 0\) under the support \(S\) that is identified by \(z_t\). To prove this, consider the minimizer \(\hat{z}(D)\). We have \(0 \in \nabla_1 f(\hat{z}, D)\), hence, we get \(0 \in J(D)\nabla^2_1 f(\hat{z}, D) + \nabla^2_1 f(\hat{z}, D)^T\). Given the support recovery, the relation \(J(D)(\nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S) + \nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S = 0\) also holds which is equivalent to \(Q(\hat{z}, J)\) under the support. To show this, given the recursion \(z_{t+1} = \Phi(z_t, D)\), we differentiate it and get \(J_{t+1} = \nabla_1 \Phi(z_t, D) J_t + \nabla_2 \Phi(z_t, D)\). Given the support recovery and fixed-point property, we can write

\[
\hat{J} = 1_S \odot (\nabla^2_1 \mathcal{L}_x(\hat{z}, D)^T J) + 1_S \odot (\nabla^2_1 \mathcal{L}_x(\hat{z}, D)^T) \\
\hat{J} - 1_S \odot \hat{J} = -\nabla^2_1 \mathcal{L}_x(\hat{z}, D)^T \odot 1_S + \nabla^2_1 \mathcal{L}_x(\hat{z}, D)^T \odot 1_S \\
0 = \hat{J}^T(\nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S) + \nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S \\
\hat{J}^+ = -\nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S(\nabla^2_1 \mathcal{L}_x(\hat{z}, D) \circ 1_S)^{-1} \tag{125}
\]

For the Jacobian corresponding to row \(i\) of the dictionary, we get

\[
\hat{J}_{(i,:)} = -(D^T S D)_{i,:}^{-1}(D_{i,:} \hat{z}^T + (D^T_{i,:} \hat{z} - x_i)J_{i}) \tag{127}
\]

on the support. Outside of the support \(S\), the Jacobian is zero. This proof is similarly provided by Malézieux et al. (2022). Hence, we can use \(\nabla^2_1 \mathcal{L}_x(\hat{z}, D) = -\hat{J}^T \nabla^2_1 \mathcal{L}_x(\hat{z}, D)\) under the support \(S\) in the following

\[
\|Q(\hat{z}, J)\|_2 = \|J^+ \nabla^2_1 \mathcal{L}_x(\hat{z}, D) + \nabla^2_1 \mathcal{L}_x(\hat{z}, D)\|_2 = \|J^+ \nabla^2_1 \mathcal{L}_x(\hat{z}, D) - J^+ \nabla^2_1 \mathcal{L}_x(\hat{z}, D)\|_2 \\
\leq (\|J_t - \hat{J}\|^T \nabla^2_1 \mathcal{L}_x(\hat{z}, D)\|_2 \leq L_1 \|J_t - \hat{J}\|_2. \tag{128}
\]

### A.5 Global backward pass proof details

We re-state and prove Theorem 4.10 as follows:

**Theorem 4.10** (Global error of gradients). *Given the convergence results from the forward pass, (Theorems 4.6 and 4.8), the errors of gradients defined in Algorithm 2 w.r.t global direction \(g^*\) (defined in (5)) satisfy

\[
\|g_{t}^{\text{ae-lasso}} - g^*\|_2 \leq \mathcal{O}(\|D - D^*\|_2^2 + \|D - D^*\|_2 + \delta^* + \hat{\delta} + M_1 \lambda \sqrt{s}) \\
\|g_{t}^{\text{ae-ls}} - g^*\|_2 \leq \mathcal{O}(\|D - D^*\|_2^2 + \|D - D^*\|_2 + \delta^* + \hat{\delta} + \hat{\delta}^*) \tag{13}
\]

*Proof.* For \(g_{t}^{\text{dec}}\), we compute the gradient in their limit assuming infinite fresh samples

\[
\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \nabla^2_1 \mathcal{L}_x(z^i_t, D) = \mathbb{E}_{x \in X} [\nabla^2_1 \mathcal{L}_x(z_t, D)] \text{ a.s.}.
\]

Similar to Theorem 4.9, we re-write the errors of gradients \(g_{t}^{\text{ae-lasso}}\) and \(g_{t}^{\text{ae-ls}}\) as following

\[
g_{t}^{\text{ae-lasso}} - g^* = \mathbb{E}_{x \in X} [Q(z^*, J_t)(z_t - z^*)] + \mathbb{E}_{x \in X} [Q_{t}^{21}(z^*)] + \mathbb{E}_{x \in X} [J_tQ_{t}^{\text{lasso-11}}(z^*)] \\
g_{t}^{\text{ae-ls}} - g^* = \mathbb{E}_{x \in X} [Q(z^*, J_t)(z_t - z^*)] + \mathbb{E}_{x \in X} [Q_{t}^{21}(z^*)] + \mathbb{E}_{x \in X} [J_tQ_{t}^{\text{ls-11}}(z^*)] \tag{129}
\]
where $Q_t^{21}(z)$, $Q_t^{\text{lasso-11}}(z)$, $Q_t^{\text{ls-11}}(z)$, and $Q(z, J)$ are defined as in Theorem 4.9. Given Assumption 4.1 and Lemma A.6, we find an upper bound on the r.h.s of the gradient errors as follows:

$$
\|g_t^{\text{ae-lasso}} - g^*\|_2 \leq \mathbb{E}_{x \in X} \left[ |(L_1\|J_t - J^*\|_2 + M_J L_{11D} D_{21D}\|D - D^*\|_2)|z_t - z^*\|_2 + (L_{21}/2)\|z_t - z^*\|_2^2 \right] + \mathbb{E}_{x \in X} \left[ M_J (L_{11/2})\|z_t - z^*\|_2^2 + M_J \|\partial h(z_t)\|_2 + L_{1D}\|D - D^*\|_2 \right] \\
\leq \mathbb{E}_{x \in X} \left[ L_1(\|J_t - J^*\|_2 + \|J_t - J^*\|_2 + M_J L_{11D} D_{21D}\|D - D^*\|_2)(\|z_t - \hat{z}\|_2 + \|\hat{z} - z^*\|_2) \right] + \mathbb{E}_{x \in X} \left[ (L_{21}/2)(\|z_t - \hat{z}\|_2^2 + \|\hat{z} - z^*\|_2^2) + L_{1D}\|D - D^*\|_2 \right]
$$

Similarly,

$$
\|g_t^{\text{ae-ls}} - g^*\|_2 \leq \mathbb{E}_{x \in X} \left[ |(L_1\|J_t - J^*\|_2 + M_J L_{11D} D_{21D}\|D - D^*\|_2)|z_t - z^*\|_2 + (L_{21}/2)\|z_t - z^*\|_2^2 \right] + \mathbb{E}_{x \in X} \left[ M_J (L_{11/2})\|z_t - z^*\|_2^2 + M_J \|\partial h(z_t)\|_2 \right] \\
\leq \mathbb{E}_{x \in X} \left[ L_1(\|J_t - J^*\|_2 + \|J_t - J^*\|_2 + M_J L_{11D} D_{21D}\|D - D^*\|_2)(\|z_t - \hat{z}\|_2 + \|\hat{z} - z^*\|_2) \right] + \mathbb{E}_{x \in X} \left[ (L_{21}/2)(\|z_t - \hat{z}\|_2^2 + \|\hat{z} - z^*\|_2^2) + L_{1D}\|D - D^*\|_2 \right]
$$

Using the convergence errors from the forward pass (Theorems 4.3 and 4.7),

$$
\|g_t^{\text{ae-lasso}} - g^*\|_2 \leq L_1 O(2\rho t + (\|D - D^*\|_2 + \hat{\delta} + \hat{\delta})\|\partial h(z_t)\|_2 + \hat{\delta} + \hat{\delta}) \\
+ L_1 O((\|D - D^*\|_2 + \hat{\delta}\)(\|D - D^*\|_2 + \hat{\delta})) + M_J L_{11D} D_{21D}\|D - D^*\|_2(\|\rho\|_1 + \|D - D^*\|_2 + \hat{\delta}) + M_J \|\partial h(z_t)\|_2
$$

Hence,

$$
\|g_t^{\text{ae-lasso}} - g^*\|_2 \leq O((\|D - D^*\|_2 + \hat{\delta})\|\partial h(z_t)\|_2 + \|D - D^*\|_2 + \|\rho\|_1 + \lambda \sqrt{s}) = O((\|D - D^*\|_2 + \|\rho\|_1 + \lambda \sqrt{s}) + M_J \lambda \sqrt{s})
$$

Similarly,

$$
\|g_t^{\text{ae-ls}} - g^*\|_2 \leq O((\|D - D^*\|_2 + \|D - D^*\|_2 + \|\rho\|_1 + \lambda \sqrt{s}) + M_J \lambda \sqrt{s})
$$

Lemma A.6 (Global bounds). From global gradient errors in Theorem 4.10, the following are satisfied

$$
\|Q_t^{21}(z^*)\|_2 \leq (L_{21}/2)\|z_t - z^*\|_2^2 \\
\|Q_t^{\text{lasso-11}}(z^*)\|_2 \leq (L_{11}/2)\|z_t - z^*\|_2^2 + L_{1D}\|D - D^*\|_2 + \|\partial h(z_t)\|_2 \\
\|Q_t^{\text{ls-11}}(z^*)\|_2 \leq (L_{11}/2)\|z_t - z^*\|_2^2 + L_{1D}\|D - D^*\|_2 \\
\|Q(z^*, J_t)\|_2 \leq L_1\|J_t - J^*\|_2 + M_J L_{11D} D_{21D}\|D - D^*\|_2.
$$

Proof. For $Q_t^{21}(z^*)$, we achieve the quadratic bound using convexity of $\nabla_1 L(x, D)$ and its domain (Assumption 3.1) and Lemma 3.3. For $Q_t^{\text{lasso-11}}(z^*)$, we add and subtract $\nabla_1 L(x^*, D)$, and use quadratic upper bound similar to Lemma A.5. At line four, we use $0 \in \nabla_1 L(x^*, D^*)$ (Lemma A.3) and assume that $z_t$
reverses the sign entries of $z^*$ (see Theorem 4.1 and Theorem 4.2).

$$
\|Q^\text{lasso-11}(z^*)\|_2 = \|\nabla_1 L_x(z_t, D) + \partial h(z_t) - \nabla_{11}^2 L_x(z^*, D)(z_t - z^*)\|_2
$$

$$
= \|\nabla_1 L_x(z_t, D) - \nabla_1 L_x(z^*, D) + \nabla_1 L_x(z^*, D) + \partial h(z_t) - \nabla_{11}^2 L_x(z^*, D)(z_t - z^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + \|\partial h(z_t) + \nabla_1 L_x(z^*, D)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + \|\nabla_1 L_x(z^*, D) - \nabla_1 L_x(z^*, D^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + L_{11D}\|D - D^*\|_2.
$$

Similarly,

$$
\|Q^\text{lasso-11}(z^*)\|_2 = \|\nabla_1 L_x(z_t, D) - \nabla_{11}^2 L_x(z^*, D)(z_t - z^*)\|_2
$$

$$
= \|\nabla_1 L_x(z_t, D) - \nabla_1 L_x(z^*, D) + \nabla_1 L_x(z^*, D) - \nabla_{11}^2 L_x(z^*, D)(z_t - z^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + \|\nabla_1 L_x(z^*, D) - \nabla_1 L_x(z^*, D^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + L_{11D}\|D - D^*\|_2.
$$

For $Q(z^*, J_t)$, from implicit function theorem, $Q(z^*, J^*) = 0$ for $D$ evaluated at $D^*$. Hence, we can use

$$
\|Q(z^*, J^*)_t\|_2 = \|J^+_t\nabla_{11}^2 L_x(z^*, D) + \nabla_{21}^2 L_x(z^*, D) - \nabla_{21}^2 L_x(z^*, D^*) + \nabla_{21}^2 L_x(z^*, D^*)\|_2
$$

$$
= \|J^+_t\nabla_{11}^2 L_x(z^*, D) - \nabla_{11}^2 L_x(z^*, D^*)\|_2 + \|\nabla_{21}^2 L_x(z^*, D) - \nabla_{21}^2 L_x(z^*, D^*)\|_2
$$

$$
= \|J^+_t\nabla_{11}^2 L_x(z^*, D) - J^+_t\nabla_{11}^2 L_x(z^*, D^*)\|_2 + \|\nabla_{21}^2 L_x(z^*, D) + \nabla_{21}^2 L_x(z^*, D^*)\|_2
$$

$$
\leq M_J L_{11D}\|D - D^*\|_2 + M_{21D}\|D - D^*\|_2
$$

$$
\leq M_J L_{11D}\|D - D^*\|_2 + M_{21D}\|D - D^*\|_2
$$

$$
\leq L_1\|J_t - J^*_t\|_2 + M_J L_{11D}L_{21D}\|D - D^*\|_2.
$$

$$
\|Q^\text{lasso-11}(z^*)\|_2 = \|\nabla_1 L_x(z_t, D) - \nabla_{11}^2 L_x(z^*, D)(z_t - z^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + \|\nabla_1 L_x(z^*, D) - \nabla_1 L_x(z^*, D^*)\|_2
$$

$$
\leq (L_{11}/2)\|z_t - z^*\|_2^2 + L_{11D}\|D - D^*\|_2.
$$

Theorem 4.11 (Dictionary learning with variable $\lambda_t$). Given Assumptions 3.3 and 3.4, suppose $D^{(l)}$ is $\mu_1$-incoherent and $(\delta_t, 2)$-close to $D^*$ with $\delta_t = \Omega^*(1/\log p)$. If $s = \Omega(\sqrt{m}), \mu = \Omega(\log m)$, learning rate is $\eta = \Omega(\frac{1}{(11-\gamma/2)^2})$, and the regularizer and step size are set according to Theorem 4.4, then for any dictionary update $l$ using $g_t^{(l)}$, with probability of at least $1 - \epsilon_{\text{supp-pres}},$

$$
\|D_{j_t}^{(l+1)} - D_{j_t}^*\|_2^2 \leq (1 - \psi)\|D_{j_t}^{(l)} - D_{j_t}^*\|_2^2
$$

where $\epsilon_{\text{supp-pres}} := \epsilon_{\text{supp-rec}} + \epsilon_{\gamma} = 2p \exp(-\frac{c^2_{\text{supp}}}{\psi}) + 2s \exp(-\frac{c^2_{\text{supp}}}{\psi_{l,0}})$.

Proof. In this proof, we study $g_t^{(l)}$, and for ease of notation we drop the superscript.

$$
g_t^{(l)} = \mathbb{E}[1_{z_{t,(j)} \neq 0}z_{t,(j)}(D^{(l)}z_T - x)] = \mathbb{E}[1_{z_{t,(j)} \neq 0}z_{t,(j)}(D^{(l)}z_T - x)] + \gamma
$$

where $\gamma = \mathbb{E}[1_{z_{t,(j)} \neq 0}z_{t,(j)}(D^{(l)}z_T - x)]$. We have the event $1_{z_{t,(j)} \neq 0} - 1_{z_{t,(j)} \neq 0} = 0$ happening with probability of $1 - \epsilon_{\text{supp-pres}}$, and $\epsilon_{\text{supp-pres}}$ decreases with decrease in $\delta_t$. Hence, $\gamma$ gets smaller. We write

$$
g_t^{(l)} = \mathbb{E}[1_{z_{t,(j)} \neq 0}z_{t,(j)}(D^{(l)}z_T - x)] + \gamma
$$

where $B_{S_t}^{(l)}$ is a diagonal matrix with $\beta_{j_t}^{(l)}$ for $j \in S$ as entries. For $j \notin S$, $1_{z_{t,(j)} \neq 0} = 0$, which results in $g_t^{(l)} = 0$. Hence, we only focus on $j \in S$ where $1_{z_{t,(j)} \neq 0} = 1$. We condition on the support and re-write the
where in the last line, we use the fact that $E[z^*_j | j \in S] = 0$ and $E[z^*_S z^*_T] = I$. Computing the expectation, we get

$$g_{T,j}^{(l)} = p_D D_j^{(l)} (1 - \beta_j^{(l)})^2 - p_D D_j^{(l)} (1 - \beta_j^{(l)}) + U_{T,j}^{(l)} + \gamma = p_D (1 - \beta_j^{(l)}) (1 - \beta_j^{(l)}) D_j^{(l)} - D_j^{(l)} + U_{T,j}^{(l)} + \gamma$$

(142)

where $U_{T,j}^{(l)} = E[D_S^{(l)} (I - B_S^{(l)}) E[z^*_S c_{T,j}^{(l)}] | S] - E[D_S^{(l)} E[z^*_S c_{T,j}^{(l)}] | S] + E[D_S^{(l)} E[(z^*_j (1 - \beta_j^{(l)}) + c_{T,j}^{(l)}) z^*_S] | S]$.

Given this gradient, we now find a bound on $U_{T,j}^{(l)}$.

$$U_{T,j}^{(l)} = E[D_S^{(l)} (I - B_S^{(l)}) E[z^*_S c_{T,j}^{(l)}] | S] - E[D_S^{(l)} E[z^*_S c_{T,j}^{(l)}] | S] + E[D_S^{(l)} E[(z^*_j (1 - \beta_j^{(l)}) + c_{T,j}^{(l)}) c_{T,j}^{(l)}] | S]$$

(143)

First, we bound $E[z^*_j c_{T,j}^{(l)} | S]$ as following.

$$E[z^*_j c_{T,j}^{(l)} | S] \leq \sum_{t=1}^{T} \alpha (1 - \alpha)^{T-t} E[2 \gamma_j^{(l)} z^*_i | S] + \frac{\mu \gamma}{\sqrt{m}} \sum_{t=1}^{T} \alpha (1 - \alpha)^{T-t} \sum_{k \neq j} E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) z^*_i | S]$$

$$+ \frac{\mu \gamma}{\sqrt{m}} \sum_{t=1}^{T} \alpha (1 - \alpha)^{T-t} \sum_{k \neq j} E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) \text{sign}(z_{t,i}) z^*_i | S] + \kappa_{T,j}^{(l)}$$

(144)

where $\kappa_{T,j}^{(l)} = E[z^*_j \kappa_{T,j}^{(l)}]$. Similar to $\kappa_{T,j}^{(l)}$, $\kappa_{T,j}^{(l)}$ decay very fast as $T$ increases. Hence, we bound the other terms. We have

$$E[\gamma_j^{(l)} z^*_i | S] \begin{cases} \leq \delta & \text{if } j \neq i \\ 0 & \text{if } j = i \end{cases}$$

(145)

$$E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) z^*_i | S] \begin{cases} \leq E_{t-1,k} & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}$$

(146)

and

$$E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) \text{sign}(z_{t,i}) z^*_i | S] \begin{cases} \leq E_{t-1,i} & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}$$

(147)

Hence, we have

$$\sum_{k \neq j} E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) z^*_i | S] \leq E_{t-1,i} \begin{cases} \leq E_{t-1,i} & \text{if } j \neq i \\ 0 & \text{if } j = i \end{cases}$$

(148)

and

$$\sum_{k \neq j} E[E_{t-1,k} \text{sign}(z^*_k - z_{t-1,k}) \text{sign}(z_{t,i}) z^*_i | S] \leq E_{t-1,i} \begin{cases} \leq E_{t-1,i} & \text{if } j \neq i \\ 0 & \text{if } j = i \end{cases}$$

(149)
Hence, for \( j \neq i \), we can write
\[
E[z_{(i)}^{*}c_{T,j}^{(l)} | S] \leq 2\delta_t + 2 \frac{\mu_t}{\sqrt{m}} \sum_{t=1}^{T} \alpha(1-\alpha)^{T-t} E_{t-1,i} + \tilde{\kappa}_{T,j}^{(l)}
\]  
(150)

where from (103), we have
\[
E_{t-1,i}(1-\alpha)^{T-t} \leq (1 + 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} (t-1)) E_{0,max} \delta_{\alpha,T-2}
\]

Moreover, with
\[
\beta_{max}^{(l)} | z_{max}^{*} | + 2\gamma_{max}^{(l)} \left( \sum_{t=1}^{T} (1-\alpha)^{T-k-1} + \sum_{t=1}^{T} 2(s-1)\kappa_{i}(1-\alpha)^{T-t} \right)
\]  
(151)

Hence, given the sparsity level, the term below is bounded by \( a_{\gamma} \) with probability of \( 1 - \epsilon_{\gamma}(l) \).
\[
\sum_{t=1}^{T} E_{t-1,i}(1-\alpha)^{T-t} \leq \sum_{t=1}^{T} (1 + 2(s-1)\alpha \frac{\mu_t}{\sqrt{m}} (t-1)) E_{0,max} \delta_{\alpha,T-2}
\]

Finally, we get
\[
E[z_{(i)}^{*}c_{T,j}^{(l)} | S] \leq \begin{cases} 2\delta_t + \frac{\mu_t}{\sqrt{m}} O(a_{\gamma}(l)) + \kappa_{T,j}^{(l)} & \text{if } j \neq i \\ \tilde{\kappa}_{T,j}^{(l)} & \text{if } j = i \end{cases}
\]  
(153)

For appropriately large \( T \), \( \kappa_{T,j}^{(l)} \) can be make small; Hence, in this case, we get
\[
\|U_{T,j}^{(l)}\|_2 \leq O(\sqrt{pp_{ij}} \delta_t \|D^{(l)}\|_2)
\]  
(154)

Now, we can re-write the gradient as
\[
g_{T,j}^{(l)} = p_j (1 - \beta_j^{(l)})(D_j^{(l)} - D_j^*) + p_j (\beta_j^{(l)} D_j^{(l)} + \frac{1}{p_j} U_{T,j}^{(l)} + \frac{1}{p_j} \gamma) = \tau(D_j^{(l)} - D_j^*) + \theta
\]  
(155)

where \( \tau = p_j (1 - \beta_j^{(l)}) \), and \( \theta = p_j (\beta_j^{(l)} D_j^{(l)} + \frac{1}{p_j} U_{T,j}^{(l)} + \frac{1}{p_j} \gamma) \). We can bound the norm of \( \theta \) as follows:
\[
\|\theta\|_2 \leq p_j \beta_j^{(l)} \|D_j^{(l)}\|_2 + \|U_{T,j}^{(l)}\|_2 + \gamma
\]  
(156)

Given \( \|D_j^{(l)}\|_2 = 1 \), and \( \beta_j^{(l)} = \langle D_j^* - D_j^{(l)}, D_j^* \rangle = \frac{1}{2} \|D_j^{(l)} - D_j^*\|_2^2 \), we modify the upper bound
\[
\|\theta\|_2 \leq \frac{1}{2} p_j \|D_j^{(l)} - D_j^*\|_2^2 + O(\sqrt{pp_{ij}} \delta_t \|D^{(l)}\|_2) + \gamma
\]  
(157)

We assume a dictionary closeness during training, i.e., \( \|D^{(l)} - D^*\|_2 \leq 2 \|D^*\|_2 \), which we prove in Lemma A.7. Given this closeness, we have
\[
\|D^{(l)}\|_2 \leq \|D^{(l)} - D^*\|_2 + \|D^*\|_2 = O(\sqrt{\frac{\mu}{m}})
\]  
(158)

Moreover, with \( \gamma \) dropping with \( \delta_t \), and for \( s = O(\sqrt{m}) \), it is reduced to
\[
\|\theta\|_2 \leq p_j \|D_j^{(l)} - D_j^*\|_2
\]  
(159)

We get
\[
\|g_{T,j}^{(l)}\|_2 \leq p_j (1 - \beta_j^{(l)}) \|D_j^{(l)} - D_j^*\|_2 + p_j \|D_j^{(l)} - D_j^*\|_2
\]
\[
\|g_{T,j}^{(l)}\|_2^2 \leq p_j^2 (2 - \beta_j^{(l)})^2 \|D_j^{(l)} - D_j^*\|_2
\]  
(160)
Using this bound, we can find a lower bound on the correlation between the gradient direction and the desired direction as follows

\[\|\mathbf{g}_{T,j}^{(l)}\|_2^2 = (p_j(1 - \beta_j^{(l)})^2\|\mathbf{D}_{j}^{(l)} - \mathbf{D}_j^*\|_2^2 + \|\theta\|_2^2 + 2p_j(1 - \beta_j^{(l)})\langle \theta, \mathbf{D}_j^{(l)} - \mathbf{D}_j^* \rangle) \]

\[2\langle \theta, \mathbf{D}_j^{(l)} - \mathbf{D}_j^* \rangle = -p_j(1 - \beta_j^{(l)})\|\mathbf{D}_j^{(l)} - \mathbf{D}_j^*\|_2^2 + \frac{1}{p_j(1 - \beta_j^{(l)})}\|\mathbf{g}_{T,j}^{(l)}\|_2^2 - \frac{1}{p_j(1 - \beta_j^{(l)})}\|\theta\|_2^2 \]

\[2\langle \mathbf{g}_{T,j}^{(l)}, \mathbf{D}_j^{(l)} - \mathbf{D}_j^* \rangle = p_j(1 - \beta_j^{(l)})\|\mathbf{D}_j^{(l)} - \mathbf{D}_j^*\|_2^2 + \frac{1}{p_j(1 - \beta_j^{(l)})}\|\mathbf{g}_{T,j}^{(l)}\|_2^2 - \frac{1}{p_j(1 - \beta_j^{(l)})}\|\theta\|_2^2 \]

(161)

Hence, using the descent property of Theorem 6 from (Arora et al., 2015), setting the learning rate to

\[\eta = \max_j \frac{1}{p_j(1 - \beta_j^{(l)})}, \quad \psi = \eta p_j(1 - \beta_j^{(l)}) - \frac{1}{p_j(1 - \beta_j^{(l)})} \leq 1 - \frac{1}{(1 - \beta_j^{(l)})^2} \]

\[\|\mathbf{D}_j^{(l+1)} - \mathbf{D}_j^*\|_2^2 \leq (1 - \psi)\|\mathbf{D}_j^{(l)} - \mathbf{D}_j^*\|_2^2 \]

(162)

**Theorem 4.12** (Dictionary learning with fixed \(\lambda_i\)). Given Assumptions 3.3 and 3.4, suppose \(\mathbf{D}^{(l)}\) is \(\mu\)-incoherent and \((\delta_i, 2)\)-close to \(\mathbf{D}^*\) with \(\delta_i = \mathcal{O}(1/\log p)\). If \(s = \mathcal{O}(\sqrt{m})\), \(\mu = \mathcal{O}(\log m)\), learning rate is

\[\eta = \mathcal{O}(\frac{p}{\sqrt{1 - \delta_i^2}}), \quad \text{and the regularizer} \quad \lambda^{\text{fixed}}, \quad \text{and step size are set according to Theorem 4.5, then for any} \]

**Proof.** Following steps similar to Theorem 4.11, we write the gradient as

\[\mathbf{g}_{T,j}^{(l)} = p_j\mathbf{D}_j^{(l)}(1 - \beta_j^{(l)})^2 - p_j\mathbf{D}_j^*(1 - \beta_j^{(l)})^2 + U_{T,j}^{(l)} + \gamma = p_j(1 - \beta_j^{(l)})\left(1 - \beta_j^{(l)}\right)\mathbf{D}_j^{(l)} - \mathbf{D}_j^*\right) + U_{T,j}^{(l)} + \gamma \]

(163)

where \(U_{T,j}^{(l)} = \mathbb{E}\left[D_{S}^{(l)}(I - B_{S}^{(l)})E[z_{(j)}^*]^{T,j} | S] - \mathbb{E}\left[D_{S}^{(l)}E[z_{(j)}^*]^{T,j} | S] + \mathbb{E}[D_{S}^{(l)}E[(z_{(j)}^*)^2]^{T,j} | S] \right.\]

Given this gradient, we now find a bound on \(U_{T,j}^{(l)}\). First, we bound \(\mathbb{E}[z_{(i)}^*]^{T,j} | S] \)

\[\mathbb{E}[z_{(i)}^*]^{T,j} | S)] \]

(164)

where we set all \(\lambda_{(i)}^{(l)}_{-1,j} = \lambda^{\text{fixed}}\) and \(\tilde{\kappa}_{T,j}^{(l)} = \mathbb{E}[z_{(i)}^*]^{T,j} | S] \)

Similar to \(\kappa_{T,j}^{(l)}\), \(\tilde{\kappa}_{T,j}^{(l)}\) decay very fast as \(T\) increases. Hence, we bound the other terms. We have

\[\mathbb{E}[\gamma_j^{(l)}z_{(i)}^* | S)] \leq \delta_i \quad \text{if } j \neq i \]

\[= 0 \quad \text{if } j = i \]

(165)

\[\mathbb{E}[\lambda^{\text{fixed}}\text{sign}(z_{(i)}^{(l)}z_{(i)}^*) | S)] = \lambda^{\text{fixed}} \quad \text{if } j = i \]

\[= 0 \quad \text{if } j \neq i \]

(166)

\[\mathbb{E}[E_{t-1,k}z_{(i)}^* | S)] \leq E_{t-1,k} \quad \text{if } k = i \]

\[= 0 \quad \text{if } k \neq i \]

(167)
Hence,
\[
\sum_{k \neq j} \mathbb{E}[E_{t-1,k} \text{sign}(z_{t(k)} - z_{t-1,(k)})z_{t(i)} | S] \leq E_{t-1,i} \begin{cases} 
E_{t-1,i} & \text{if } j \neq i \\
0 & \text{if } j = i 
\end{cases} 
\] (168)

Hence, for \( j \neq i \), we can write
\[
\mathbb{E}[z_{(i)}^t \zeta_j^{(l)} | S] \leq \delta_t + \frac{Ma}{\sqrt{m}} \left( \sum_{i=1}^T \alpha(1-\alpha)^{i-t}(E_{0,i} + E_{t-1,i}) + \tilde{\kappa}_T^{(l)} \right) 
\] (169)

We have
\[
E_{t-1,i}(1-\alpha)^{T-t} \leq (1 + 2(s-1)\alpha \frac{Ma}{\sqrt{m}} (t-1))E_{0,max} \delta_{a,T-2}
\]
\[
+ (\beta_{\text{max}}^{(l)} | z_{\text{max}}^* | + 2\kappa_{\text{max}}^{(l)}) \left( \sum_{k=1}^{t-1} \alpha(1-\alpha)^{T-k-1} + \sum_{k=1}^T 2(s-1)\kappa_i(1-\alpha)^{T-t} \right) 
\] (170)

Hence, given the sparsity level, the term below is bounded by \( a_{\gamma} \), with probability of \( 1 - \epsilon_{\gamma}^{(l)} \).
\[
\sum_{t=1}^T E_{t-1,i}(1-\alpha)^{T-t} \leq \sum_{t=1}^T (1 + 2(s-1)\alpha \frac{Ma}{\sqrt{m}} (t-1))E_{0,max} \delta_{a,T-2}
\]
\[
+ (\beta_{\text{max}}^{(l)} | z_{\text{max}}^* | + 2\kappa_{\text{max}}^{(l)}) \left( \sum_{t=1}^{T-1} \alpha(1-\alpha)^{T-k-1} + \sum_{t=1}^T 2(s-1)\kappa_i(1-\alpha)^{T-t} \right) 
\] (171)

Finally, we get
\[
\mathbb{E}[z_{(i)}^t \zeta_j^{(l)} | S] \leq \begin{cases} 
\delta_t + \frac{Ma}{\sqrt{m}} \mathcal{O}(a_{\gamma}^{(l)}) + \tilde{\kappa}_T^{(l)} \zeta_j^{(l)} & \text{if } j \neq i \\
\lambda_{\text{fixed}}^{(l)} + \tilde{\kappa}_T^{(l)} & \text{if } j = i 
\end{cases} 
\] (172)

For appropriately large \( T \), \( \tilde{\kappa}_T^{(l)} \) can be made small; Hence, in this case, we get
\[
\|U_T^{(l)}\|_2 \leq \mathcal{O}(\sqrt{p_{\text{fixed}}^l} \delta_l \|D^{(l)}\|_2 + p_j \lambda_{\text{fixed}}^{(l)}) 
\] (173)

Now, we can re-write the gradient as
\[
g_T^{(l)} = p_j(1 - \beta_j^{(l)})(D_j^{(l)} - D_j^*) + p_j( -\beta_j^{(l)}D_j^{(l)} + \frac{1}{p_j} U_{T,j}^l + \frac{1}{p_j} ) \gamma) = \tau(D_j^{(l)} - D_j^*) + \theta 
\] (174)

where \( \tau = p_j(1 - \beta_j^{(l)}) \), and \( \theta = p_j( -\beta_j^{(l)}D_j^{(l)} + \frac{1}{p_j} U_{T,j}^l + \frac{1}{p_j} \gamma) \). We can bound the norm of \( \theta \) as follows:
\[
\|\theta\|_2 \leq p_j \beta_j^{(l)} \|D_j^{(l)}\|_2^2 + \|U_{T,j}^l\|_2 + \gamma 
\] (175)

Given \( \|D_j^{(l)}\|_2^2 = 1 \), and \( \beta_j^{(l)} = \langle D_j^* - D_j^{(l)}, D_j^* \rangle \left[ \frac{1}{2} \|D_j^{(l)} - D_j^*\|_2^2 \right] \), we modify the upper bound
\[
\|\theta\|_2 \leq \frac{1}{2} p_j \|D_j^{(l)} - D_j^*\|_2^2 + \mathcal{O}(\sqrt{p_{\text{fixed}}^l} \delta_l \|D^{(l)}\|_2 + p_j \lambda_{\text{fixed}}) + \gamma 
\] (176)

We assume a dictionary closeness during training, i.e., \( \|D^{(l)} - D^*\|_2 \leq 2\|D^*\|_2 \), which we prove in Lemma A.7. Given this closeness, we have
\[
\|D^{(l)}\|_2 \leq \|D^{(l)} - D^*\|_2 + \|D^*\|_2 = \mathcal{O} \left( \sqrt{\frac{p}{m}} \right) 
\] (177)

Moreover, with \( \gamma \) dropping with \( \delta_l \), and for \( s = \mathcal{O}(\sqrt{m}) \), it is reduced to
\[
\|\theta\|_2 \leq p_j (\|D_j^{(l)} - D_j^*\|_2 + \lambda_{\text{fixed}}) 
\] (178)
We get
\[
\|g_{T,j}^{(l)}\|_2^2 \leq p_j (1 - \beta_j^{(l)})\|D_j^{(l)} - D_j^*\|_2^2 + p_j (\|D_j^{(l)} - D_j^*\|_2 + \lambda^{\text{fixed}})
\]
\[
\|g_{T,j}^{(l)}\|_2^2 \leq 2p_j^2 (2 - \beta_j^{(l)})^2 \|D_j^{(l)} - D_j^*\|_2^2 + 2p_j^2 \lambda^{\text{fixed2}}
\]  
(179)

Using this bound, we can find a lower bound on the correlation between the gradient direction and the desired direction as follows
\[
\|g_{T,j}^{(l)}\|_2^2 = (p_j(1 - \beta_j^{(l)}))^2\|D_j^{(l)} - D_j^*\|_2^2 + \|\theta\|_2^2 + 2p_j(1 - \beta_j^{(l)})\langle \theta, D_j^{(l)} - D_j^* \rangle
\]
\[
2\langle \theta, D_j^{(l)} - D_j^* \rangle = -p_j(1 - \beta_j^{(l)})\|D_j^{(l)} - D_j^*\|_2^2 + \frac{1}{p_j(1 - \beta_j^{(l)})}\|g_{T,j}^{(l)}\|_2^2 - \frac{1}{p_j(1 - \beta_j^{(l)})}\|\theta\|_2^2
\]
\[
2\langle g_{T,j}^{(l)}, D_j^{(l)} - D_j^* \rangle = p_j(1 - \beta_j^{(l)})\|D_j^{(l)} - D_j^*\|_2^2 + \frac{1}{p_j(1 - \beta_j^{(l)})}\|g_{T,j}^{(l)}\|_2^2 - \frac{1}{p_j(1 - \beta_j^{(l)})}\|\theta\|_2^2
\]
\[
\geq (p_j(1 - \beta_j^{(l)}) - 2p_j \frac{1}{1 - \beta_j^{(l)}}\|D_j^{(l)} - D_j^*\|_2^2 + \frac{1}{p_j(1 - \beta_j^{(l)})}\|g_{T,j}^{(l)}\|_2^2 - \frac{2p_j}{(1 - \beta_j^{(l)})}\lambda^{\text{fixed2}}
\]  
(180)

Hence, using the descent property of Theorem 6 from (Arora et al., 2015), setting the learning rate to \( \eta = \max \frac{1}{p_j(1 - \beta_j^{(l)})} \), and \( \psi = \eta(p_j(1 - \beta_j^{(l)}) - 2p_j \frac{1}{1 - \beta_j^{(l)}}) \)
\[
\|D_j^{(l+1)} - D_j^*\|_2^2 \leq (1 - \psi)\|D_j^{(l)} - D_j^*\|_2^2 + \epsilon_\lambda
\]  
(181)

where \( \epsilon_\lambda := \eta\frac{2p_j}{(1 - \beta_j^{(l)})}\lambda^{\text{fixed2}} \)

**Lemma A.7 (Dictionary maintains closeness).** Suppose \( D_j^{(l)} \) has \((\delta_l, 2)\)-closeness to \( D^* \) where \( \delta_l = O^*(1/\log m) \), then with probability of \( 1 - \epsilon_{\text{supp-pres}}^{(l)} \), we have \( \|D_j^{(l+1)} - D^*\|_2 \leq 2\|D^*\|_2 \) when using \( g_{T,j}^{\text{desc}} \) and the network parameters set by Theorem 4.4.

**Proof.** Given the dictionary update
\[
D_j^{(l+1)} - D_j^* = D_j^{(l)} - D_j^* - \eta g_{T,j}^{(l)}
\]  
(182)

Then, with probability at least \( 1 - \epsilon_{\text{supp-pres}}^{(l)} \), we have the gradient
\[
g_{T,j}^{(l)} = p_j(1 - \beta_j^{(l)})(D_j^{(l)} - D_j^*) + p_j(-\beta_j^{(l)}D_j^{(l)} + 1 - \gamma)
\]  
(183)

which we substitute in the dictionary update as below
\[
D_j^{(l+1)} - D_j^* = (1 - \eta(p_j(1 - \beta_j^{(l)})))(D_j^{(l)} - D_j^*) + \eta p_j \beta_j^{(l)} D_j^{(l)} - \eta U_{T,j}^{(l)} - \eta\gamma)
\]  
(184)

writing the update in matrix form
\[
D_j^{(l+1)} - D^* = (D_j^{(l)} - D^*)\diag(1 - \eta(p_j(1 - \beta_j^{(l)}))) + \eta D_j^{(l)}\diag(p_j \beta_j^{(l)}) - \eta D_j^{(l)}F + \eta D^*H - \eta\gamma)
\]  
(185)

where \( F_{(ij,j)} = p_j \mathbb{E}[1 - \beta_j^{(l)}]z_{(ij)}^{(c)} | S \) + \( p_j \mathbb{E}[z_{(ij)}^{(c)} | S] \) and \( H_{(ij)} = p_j \mathbb{E}[z_{(ij)}^{(c)} | S]. \)

Given, the bound \( \|U_{T,j}^{(l)}\|_2 \leq O(\sqrt{p_j}\delta_l \|D_j^{(l)}\|_2) \) from before, we get \( \|F\|_F \leq O(pp_j\delta_l) \) and \( \|H\|_F \leq O(pp_j\delta_l). \)

Hence,
\[
\|D_j^{(l)}F + D^*H\|_2 \leq \|D_j^{(l)}\|_2\|F\|_F + \|D^*\|_2\|H\|_F = O(pp_j\delta_l\|D^*\|_2) = O(\frac{s^2}{p\log m})\|D^*\|_2
\]  
(186)

Using the maintained closeness at update \( l \), we bound the terms in the dictionary update one by one below
\[
\|(D_j^{(l)} - D^*)\diag(1 - \eta(p_j(1 - \beta_j^{(l)})))\| \leq (1 - \min \eta p_j(1 - \beta_j^{(l)}))\|D_j^{(l)} - D^*\|_2 \leq 2(1 - \Omega(\eta s/p))\|D^*\|_2
\]  
(187)
\|D(l)\text{diag}(p_j\beta_j(l))\|_2 \leq \max_j p_j \delta_j^2 \|D(l) - D^* + D^*\|_2 \leq o(s/p)\|D^*\|_2 \tag{188}

Given the bounds above, the dictionary update can be bounded as following

\|D(l+1) - D^*\|_2 \leq 2(1 - \Omega(\eta s/p))\|D^*\|_2 + o(\eta s/p)\|D^*\|_2 + O(\eta s^2 p \log m)\|D^*\|_2 + \eta \gamma \leq 2\|D^*\|_2 \tag{189}

A.6 Interpretability

Theorem 5.1 (Interpretable unrolled network). Consider the dictionary learning optimization of the form

\min Z, D \frac{1}{2}\|X - DZ\|_F^2 + \lambda\|Z\|_1 + \omega/2\|D\|_F^2, \text{ where } X = [x^1, x^2, \ldots, x^n] \in \mathbb{R}^{m \times n} \text{ and } Z = [z^1, z^2, \ldots, z^n] \in \mathbb{R}^{p \times n}. \text{ Let } \hat{Z} \text{ be the given converged sparse codes, then stationary points of the problem w.r.t the network weights (dictionary) follows } D = XG^{-1}\hat{Z}^T, \text{ where we denote } G := (\hat{Z}^T\hat{Z} + \omega I)\text{a}.

Proof. For all stationary points, the objective gradient is 0 with respect to the dictionary, i.e.,

0 = (X - D\hat{Z})\hat{Z}^T + \omega \hat{D} \tag{190}

where \(\hat{D}\) is the learned dictionary at convergence. Re-arranging the terms, we get

\hat{D} = X\hat{Z}^T(\hat{Z}\hat{Z}^T + \omega I)^{-1} \tag{191}

Using the relation \(A^T(AA^T + \omega I)^{-1} = (A^TA + \omega I)^{-1}A^T\), we can re-write the solution as

\hat{D} = XG^{-1}\hat{Z}^T \tag{192}

where we denote \(G := (\hat{Z}^T\hat{Z} + \omega I)\text{a}\).
Dataset  We generated $n = 10,000$ samples following (2). We sampled $D^* \in \mathbb{R}^{50 \times 100}$ from zero-mean Gaussian distribution, and normalized the columns. The codes are 5-sparse with their support uniformly chosen at random and their amplitudes are sampled from Uniform$(1, 2)$.

Training  We let $T = 200$, $\lambda = 0.2$, and $\alpha = 0.2$. The dictionary is initialized to $D = D^* + \tau_B B$ with $B \sim \mathcal{N}(0, \frac{1}{m}I)$. For Figures 2, 3 and 4a, we set $\tau_B \approx 0.55/\log m$. Figure 11 shows the sparse code estimates from one example given this initialized dictionary; this is to highlight that a) the initial dictionary is not very close to the ground-truth dictionary, and b) our algorithm is able to successfully perform dictionary learning and recover the support by the end of training in spite of a failed exact recovery of the support.

For Figures 4b and 4c, we chose much larger noise level, $\tau_B \approx 2.8/\log m$. The network is trained for 600 epochs with full-batch gradient descent using Adam optimizer (Kingma & Ba, 2014) with learning rate of $10^{-3}$ and $\epsilon = 10^{-8}$. The learned dictionary is evaluated based on the error $\|D^*-D\|_2/\|D^*\|_2$. The results and conclusion were consistent across various realizations of the dataset and across various optimizers. Hence, in the main paper, the figures visualize results of one realization.

Noisy measurements  We repeated the dictionary learning experiments shown in Figures 4b and 4c where the measurements $x$ are corrupted by zero-mean Gaussian noise such that the SNR is approximately 12 dB. Accordingly, we set $\lambda = 0.3$. Figure 12 shows the results for both noisy and noiseless scenarios.

Stochastic Dictionary Learning  In addition to the full-batch gradient descent results in the main paper, we repeated the experiments in Figure 4c using batch size of 4, 16 and 64. We observed (Figure 13) that in all scenarios PUDLE is able to learn a good estimate of the ground-truth dictionary, and $g_l^{ae}$ is superior to the other two. We note that for lower batch-size, there will be more gradient updates in one epoch, hence, the algorithm converges in lower number of epochs.

Effect of learning rate  We performed the experiments in Figure 4c for various learning rate of $10^{-4}$, $10^{-3}$, and $10^{-2}$. This shows the robustness of the gradient-based dictionary learning against learning rate. Figure 14
demonstrates such results where PUDLE successfully converges to the neighbourhood of the ground-truth dictionary: Regardless of the learning rate, $g_{ae-ls}^*$ converges to a closer neighbourhood than the other two gradients. Overall, smaller the learning rate, more epochs is needed to reach convergence.

![Figure 14: Dictionary learning for various learning rates when $T = 100$.](image)

Dictionary initialization  We conducted similar experiments to Figures 4b and 4c. We let $n = 10,000$, $m = 100$, and $p = 100$. We generated an orthogonal $D^*$. The sparse codes $z^*$ are 5-sparse and their amplitudes are drawn from sub-Gaussian $\mathcal{N}(0, 1)$. We set $\lambda = 0.05$, and $\alpha = 0.2$. We used the pairwise method proposed by Arora et al. (2015) to initialize the dictionary. This close initialization resulted in a dictionary that provides support recovery prior training. Figure 15 shows successful dictionary learning where $g_{ae-ls}^*$ converges to a closer neighbourhood of $D^*$ than the other two gradients. We used linear sum assignment optimization (i.e., `scipy.optimize.linear_sum_assignment`) to find the correct column permutations before computing the dictionary distance error.

![Figure 15: Dictionary learning when $D$ is initialized using the pairwise method Arora et al. (2015).](image)

C.2 Dictionary learning

Dataset  We generated $n = 50,000$ samples following (2). We let $m = 1000$ and $p = 1500$, and sample $D^*$ from zero-mean Gaussian distribution, and then normalized the columns. The sparse codes $z^t$ are 10, 20, 40-sparse, where their supports are chosen uniformly at random and amplitudes are sampled from Uniform(1, 2).

Training  The dictionary is initialized to $D = D^* + \tau_B B$ with $B \sim \mathcal{N}(0, \frac{1}{m} I)$ where $\tau_B \approx 1/\log m$. We let $\lambda = 0.2$, and $\alpha = 0.2$, and $T = 100$. The network is trained for 1,000 iterative updates with batch-size of 50 using Adam (Kingma & Ba, 2014) with learning rate of $10^{-3}$ and $\epsilon = 10^{-3}$. For decay method, $\nu$ is decreased in value by 0.005 every 100 update iterations. Each filter is normalized after every update. The learned dictionary is evaluated based on the relative error $\|D - D^*\|_2/\|D^*\|_2$. 

48
C.3 Image denoising

Training We trained PUDLE where the dictionary is convolutional with 64 filters of size $9 \times 9$ and strides of 4. The encoder unrolls for $T = 15$, and the step size is set to $\alpha = 0.1$. Unlike the theoretical analysis where full-batch gradient descent is studied, the network is trained stochastically with Adam optimizer (Kingma & Ba, 2014) with a learning rate of $10^{-4}$ and $\epsilon = 10^{-3}$ for 250 epochs. At every training iteration, a random $129 \times 129$ patch is cropped and a zero-mean Gaussian noise with a standard deviation of 25 is added. We utilize random horizontal and vertical flip for augmentation. We report results in terms of the peak signal-to-noise ratio (PSNR). The standard deviation of the test PSNR across multiple noise realizations was lower than 0.02 dB for all the methods. Hence, we only reported the mean PSNR of the test set.

C.4 Interpretable sparse coding and dictionary learning

We focused on digits of $\{0, 1, 2, 3, 4\}$ of MNIST. We set $T = 15$, $\lambda = 0.7$, and $\alpha = 1$. The dictionary dimensions are $m = 784$ and $p = 500$. We trained the network for 200 epochs using Adam optimizer with a learning rate of $10^{-4}$ and batch size of 32. For construction of $G$, $\omega$ is set to 0.001. For Figure 8, we computed the image contributions using 6,000 randomly chosen training images. The Gram matrix used in Figure 9, is constructed by 6,000 training examples, and the reconstruction is from the 200 most contributed training images. 

Figure 16: Dictionary learning convergence using $\mathcal{g}^{\text{ae}-\text{ls}}$ compared to NOODL and SPORCO.