Appendix to *DessiLBI for deep learning: structural sparsity via differential inclusion paths*

A Proof of Theorem 1

First of all, we reformulate Eq. (8) into an equivalent form. Without loss of generality, consider $\Omega = \Omega_1$ in the sequel.

Denote $R(P) := \Omega(\Gamma)$, then Eq. (8) can be rewritten as, DessiLBI

$$P_{k+1} = \mathrm{Prox}_{\kappa R}(P_k + \kappa(p_k - \alpha\nabla\bar{\mathcal{L}}(P_k))), \tag{17a}$$

$$p_{k+1} = p_k - \kappa^{-1}(P_{k+1} - P_k + \kappa\alpha\nabla\bar{\mathcal{L}}(P_k)), \tag{17b}$$

where $p_k = [0, g_k]^T \in \partial R(P_k)$ and $g_k \in \partial\Omega(\Gamma_k)$. Thus DessiLBI is equivalent to the following iterations,

$$W_{k+1} = W_k - \kappa\alpha\nabla_W\bar{\mathcal{L}}(W_k, \Gamma_k), \tag{18a}$$

$$\Gamma_{k+1} = \mathrm{Prox}_{\kappa\Omega}(\Gamma_k + \kappa(g_k - \alpha\nabla_\Gamma\bar{\mathcal{L}}(W_k, \Gamma_k))), \tag{18b}$$

$$g_{k+1} = g_k - \kappa^{-1}(\Gamma_{k+1} - \Gamma_k + \kappa\alpha\cdot\nabla_\Gamma\bar{\mathcal{L}}(W_k, \Gamma_k)). \tag{18c}$$

Exploiting the equivalent reformulation (18a-18c), one can establish the global convergence of (W_k, Γ_k, g_k) based on the Kurdyka-Łojasiewicz framework. In this section, the following extended version of Theorem 1 is actually proved.

Theorem 2. *[Global Convergence of DessiLBI] Suppose that Assumption 1 holds. Let (W_k, Γ_k, g_k) be the sequence generated by DessiLBI (Eq. (18a-18c)) with a finite initialization. If*

$$0 < \alpha_k = \alpha < \frac{2}{\kappa(Lip + \nu^{-1})},$$

then (W_k, Γ_k, g_k) converges to a critical point of F. Moreover, $\{(W_k, \Gamma_k)\}$ converges to a stationary point of $\bar{\mathcal{L}}$ defined in Eq. 4, and $\{W^k\}$ converges to a stationary point of $\widehat{\mathcal{L}}_n(W)$.

A.1 Kurdyka-Łojasiewicz Property

To introduce the definition of the Kurdyka-Łojasiewicz (KL) property, we need some notions and notations from variational analysis, which can be found in (Rockafellar & Wets, 1998).

The notion of subdifferential plays a central role in the following definitions. For each $\mathbf{x} \in \mathrm{dom}(h) := \{\mathbf{x} \in \mathbb{R}^p : h(\mathbf{x}) < +\infty\}$, the *Fréchet subdifferential* of h at $\mathbf{x}$, written $\widehat{\partial}h(\mathbf{x})$, is the set of vectors $\mathbf{v} \in \mathbb{R}^p$ which satisfy

$$\lim \inf_{\mathbf{y}\neq\mathbf{x}, \mathbf{y}\to\mathbf{x}} \frac{h(\mathbf{y}) - h(\mathbf{x}) - \langle \mathbf{v}, \mathbf{y} - \mathbf{x}\rangle}{\|\mathbf{x} - \mathbf{y}\|} \geq 0.$$

When $\mathbf{x} \notin \mathrm{dom}(h)$, we set $\widehat{\partial}h(\mathbf{x}) = \varnothing$. The *limiting-subdifferential* (or simply *subdifferential*) of h introduced in (Mordukhovich, 2006), written $\partial h(\mathbf{x})$ at $\mathbf{x} \in \mathrm{dom}(h)$, is defined by

$$\partial h(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^p : \exists \mathbf{x}^k \to \mathbf{x},\ h(\mathbf{x}^k) \to h(\mathbf{x}),\ \mathbf{v}^k \in \widehat{\partial}h(\mathbf{x}^k) \to \mathbf{v}\}. \tag{19}$$

A necessary (but not sufficient) condition for $\mathbf{x} \in \mathbb{R}^p$ to be a minimizer of h is $\mathbf{0} \in \partial h(\mathbf{x})$. A point that satisfies this inclusion is called *limiting-critical* or simply *critical*. The distance between a point $\mathbf{x}$ to a subset $\mathcal{S}$ of $\mathbb{R}^p$, written $\mathrm{dist}(\mathbf{x}, \mathcal{S})$, is defined by $\mathrm{dist}(\mathbf{x}, \mathcal{S}) = \inf\{\|\mathbf{x} - \mathbf{s}\| : \mathbf{s} \in \mathcal{S}\}$, where $\|\cdot\|$ represents the Euclidean norm.

Let $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ be an extended-real-valued function (respectively, $h : \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ be a point-to-set mapping), its *graph* is defined by

$$\mathrm{Graph}(h) := \{(\mathbf{x}, y) \in \mathbb{R}^p \times \mathbb{R} : y = h(\mathbf{x})\},$$
$$(\text{resp. } \mathrm{Graph}(h) := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^p \times \mathbb{R}^q : \mathbf{y} \in h(\mathbf{x})\}),$$

and its domain by $\mathrm{dom}(h) := \{\mathbf{x} \in \mathbb{R}^p : h(\mathbf{x}) < +\infty\}$ (resp. $\mathrm{dom}(h) := \{\mathbf{x} \in \mathbb{R}^p : h(\mathbf{x}) \neq \varnothing\}$). When h is a proper function, i.e., when $\mathrm{dom}(h) \neq \varnothing$, the set of its global minimizers (possibly empty) is denoted by

$$\arg\min h := \{\mathbf{x} \in \mathbb{R}^p : h(\mathbf{x}) = \inf h\}.$$

The KL property (Łojasiewicz, 1963; 1993; Kurdyka, 1998; Bolte et al., 2007a;b) plays a central role in the convergence analysis of nonconvex algorithms (Attouch et al., 2013; Wang et al., 2019). The following definition is adopted from (Bolte et al., 2007b).

Definition 1. *[Kurdyka-Łojasiewicz property] A function h is said to have the Kurdyka-Łojasiewicz (KL) property at $\bar{u} \in \mathrm{dom}(\partial h) := \{v \in \mathbb{R}^n | \partial h(v) \neq \emptyset\}$, if there exists a constant $\eta \in (0, \infty)$, a neighborhood $\mathcal{N}$ of $\bar{u}$ and a function $\phi : [0, \eta) \to \mathbb{R}_+$, which is a concave function that is continuous at 0 and satisfies $\phi(0) = 0$, $\phi \in \mathcal{C}^1((0, \eta))$, i.e., ϕ is continuous*

differentiable on $(0,\eta)$*, and* $\phi'(s) > 0$ *for all* $s \in (0,\eta)$*, such that for all* $u \in \mathcal{N} \cap \{u \in \mathbb{R}^n | h(\bar{u}) < h(u) < h(\bar{u}) + \eta\}$*, the following inequality holds*

$$\phi'(h(u) - h(\bar{u})) \cdot \mathrm{dist}(0, \partial h(u)) \geq 1. \tag{20}$$

If h *satisfies the KL property at each point of* $\mathrm{dom}(\partial h)$*,* h *is called a KL function.*

KL functions include real analytic functions, semialgebraic functions, tame functions defined in some o-minimal structures (Kurdyka, 1998; Bolte et al., 2007b), continuous subanalytic functions (Bolte et al., 2007a) and locally strongly convex functions. In the following, we provide some important examples that satisfy the Kurdyka-Łojasiewicz property.

Definition 2. *[Real analytic] A function* h *with domain an open set* $U \subset \mathbb{R}$ *and range the set of either all real or complex numbers, is said to be* ***real analytic*** *at* u *if the function* h *may be represented by a convergent power series on some interval of positive radius centered at* u*:* $h(x) = \sum_{j=0}^{\infty} \alpha_j (x-u)^j$*, for some* $\{\alpha_j\} \subset \mathbb{R}$*. The function is said to be* ***real analytic*** *on* $V \subset U$ *if it is real analytic at each* $u \in V$ *(Krantz & Parks, 2002, Definition 1.1.5). The real analytic function* f *over* $\mathbb{R}^p$ *for some positive integer* $p > 1$ *can be defined similarly.*

According to (Krantz & Parks, 2002), typical real analytic functions include polynomials, exponential functions, and the logarithm, trigonometric and power functions on any open set of their domains. One can verify whether a multivariable real function $h(\mathbf{x})$ *on* $\mathbb{R}^p$ *is analytic by checking the analyticity of* $g(t) := h(\mathbf{x} + t\mathbf{y})$ *for any* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$*.*

Definition 3. *[Semialgebraic]*

(a) A set $\mathcal{D} \subset \mathbb{R}^p$ *is called semialgebraic (Bochnak et al., 1998) if it can be represented as*

$$\mathcal{D} = \bigcup_{i=1}^{s} \bigcap_{j=1}^{t} \{\mathbf{x} \in \mathbb{R}^p : P_{ij}(\mathbf{x}) = 0, Q_{ij}(\mathbf{x}) > 0\},$$

where P_{ij}, Q_{ij} *are real polynomial functions for* $1 \leq i \leq s, 1 \leq j \leq t$*.*

(b) A function $h : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ *(resp. a point-to-set mapping* $h : \mathbb{R}^p \rightrightarrows \mathbb{R}^q$*) is called semialgebraic if its graph* $\mathrm{Graph}(h)$ *is semialgebraic.*

According to (Łojasiewicz, 1965; Bochnak et al., 1998) and (Shiota, 1997, I.2.9, page 52), the class of semialgebraic sets are stable under the operation of finite union, finite intersection, Cartesian product or complementation. Some typical examples include polynomial functions, the indicator function of a semialgebraic set, and the Euclidean norm (Bochnak et al., 1998, page 26).

A.2 KL Property in Deep Learning and Proof of Corollary 1

In the following, we consider the deep neural network training problem. Consider a l-layer feedforward neural network including $l-1$ hidden layers of the neural network. Particularly, let d_i be the number of hidden units in the i-th hidden layer for $i = 1, \ldots, l-1$. Let d_0 and d_l be the number of units of input and output layers, respectively. Let $W^i \in \mathbb{R}^{d_i \times d_{i-1}}$ be the weight matrix between the $(i-1)$-th layer and the i-th layer for any $i = 1, \ldots l$[6].

According to Theorem 2, one major condition is to verify the introduced Lyapunov function F defined in (11) satisfies the Kurdyka-Łojasiewicz property. For this purpose, we need an extension of semialgebraic set, called the *o-minimal structure* (see, for instance (Coste, 1999), (van den Dries, 1986), (Kurdyka, 1998), (Bolte et al., 2007b)). The following definition is from (Bolte et al., 2007b).

Definition 4. *[o-minimal structure] An o-minimal structure on* $(\mathbb{R}, +, \cdot)$ *is a sequence of boolean algebras* $\mathcal{O}_n$ *of "definable" subsets of* $\mathbb{R}^n$*, such that for each* $n \in \mathbb{N}$

(i) if A *belongs to* $\mathcal{O}_n$*, then* $A \times \mathbb{R}$ *and* $\mathbb{R} \times A$ *belong to* $\mathcal{O}_{n+1}$*;*

(ii) if $\Pi : \mathbb{R}^{n+1} \to \mathbb{R}^n$ *is the canonical projection onto* $\mathbb{R}^n$*, then for any* A *in* $\mathcal{O}_{n+1}$*, the set* $\Pi(A)$ *belongs to* $\mathcal{O}_n$*;*

(iii) $\mathcal{O}_n$ *contains the family of algebraic subsets of* $\mathbb{R}^n$*, that is, every set of the form*

$$\{x \in \mathbb{R}^n : p(x) = 0\},$$

where $p : \mathbb{R}^n \to \mathbb{R}$ *is a polynomial function.*

[6] To simplify notations, we regard the input and output layers as the 0-th and the l-th layers, respectively, and absorb the bias of each layer into W^i.

(iv) the elements of $\mathcal{O}_1$ are exactly finite unions of intervals and points.

Based on the definition of o-minimal structure, we can show the definition of the *definable function*.

Definition 5. *[Definable function] Given an o-minimal structure $\mathcal{O}$ (over $(\mathbb{R}, +, \cdot)$), a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be definable in $\mathcal{O}$ if its graph belongs to $\mathcal{O}_{n+1}$.*

According to (van den Dries & Miller, 1996; Bolte et al., 2007b), there are some important facts of the o-minimal structure, shown as follows.

(i) The collection of *semialgebraic* sets is an o-minimal structure. Recall the semialgebraic sets are Bollean combinations of sets of the form
$$\{x \in \mathbb{R}^n : p(x) = 0, q_1(x) < 0, \ldots, q_m(x) < 0\},$$
where p and q_i's are polynomial functions in $\mathbb{R}^n$.

(ii) There exists an o-minimal structure that contains the sets of the form
$$\{(x, t) \in [-1, 1]^n \times \mathbb{R} : f(x) = t\}$$
where f is real-analytic around $[-1, 1]^n$.

(iii) There exists an o-minimal structure that contains simultaneously the graph of the exponential function $\mathbb{R} \ni x \mapsto \exp(x)$ and all semialgebraic sets.

(iv) The o-minimal structure is stable under the sum, composition, the inf-convolution and several other classical operations of analysis.

The Kurdyka-Łojasiewicz property for the smooth definable function and non-smooth definable function were established in (Kurdyka, 1998, Theorem 1) and (Bolte et al., 2007b, Theorem 14), respectively. Now we are ready to present the proof of Corollary 1.

Proof. [Proof of Corollary 1] To justify this corollary, we only need to verify the associated Lyapunov function F satisfies Kurdyka-Łojasiewicz inequality. In this case and by (12), F can be rewritten as follows
$$F(\mathcal{W}, \Gamma, \mathcal{G}) = \alpha \left(\widehat{\mathcal{L}}_n(W, \Gamma) + \frac{1}{2\nu} \|W - \Gamma\|^2 \right) + \Omega(\Gamma) + \Omega^*(g) - \langle \Gamma, g \rangle.$$
Because ℓ and σ_i's are definable by assumptions, then $\widehat{\mathcal{L}}_n(W, \Gamma)$ are definable as compositions of definable functions. Moreover, according to (Krantz & Parks, 2002), $\|W - \Gamma\|^2$ and $\langle \Gamma, g \rangle$ are semi-algebraic and thus definable. Since the group Lasso $\Omega(\Gamma) = \sum_g \|\Gamma\|_2$ is the composition of ℓ_2 and ℓ_1 norms, and the conjugate of group Lasso penalty is the maximum of group ℓ_2-norm, *i.e.* $\Omega^*(\Gamma) = \max_g \|\Gamma_g\|_2$, where the ℓ_2, ℓ_1, and ℓ_∞ norms are definable, hence the group Lasso and its conjugate are definable as compositions of definable functions. Therefore, F is definable and hence satisfies Kurdyka-Łojasiewicz inequality by (Kurdyka, 1998, Theorem 1).

The verifications of other cases listed in assumptions can be found in the proof of (Zeng et al., 2019a, Proposition 1). This finishes the proof of this corollary. □

A.3 Proof of Theorem 2

Our analysis is mainly motivated by a recent paper (Benning et al., 2017), as well as the influential work (Attouch et al., 2013). According to Lemma 2.6 in (Attouch et al., 2013), there are mainly four ingredients in the analysis, that is, the *sufficient descent property*, *relative error property*, *continuity property* of the generated sequence and the *Kurdyka-Łojasiewicz property* of the function. More specifically, we first establish the *sufficient descent property* of the generated sequence via exploiting the Lyapunov function F (see, (11)) in Lemma A.4 in Section A.4, and then show the *relative error property* of the sequence in Lemma A.5 in Section A.5. The *continuity property* is guaranteed by the continuity of $\bar{\mathcal{L}}(W, \Gamma)$ and the relation $\lim_{k\to\infty} B_\Omega^{g_k}(\Gamma_{k+1}, \Gamma_k) = 0$ established in Lemma 1(i) in Section A.4. Thus, together with the Kurdyka-Łojasiewicz assumption of F, we establish the global convergence of SLBI following by (Attouch et al., 2013, Lemma 2.6).

Let $(\bar{W},\bar{\Gamma},\bar{g})$ be a critical point of F, then the following holds

$$\begin{aligned}&\partial_W F(\bar{W},\bar{\Gamma},\bar{g}) = \alpha(\nabla\widehat{\mathcal{L}}_n(\bar{W}) + \nu^{-1}(\bar{W}-\bar{\Gamma})) = 0,\\ &\partial_\Gamma F(\bar{W},\bar{\Gamma},\bar{g}) = \alpha\nu^{-1}(\bar{\Gamma}-\bar{W}) + \partial\Omega(\bar{\Gamma}) - \bar{g} \ni 0,\\ &\partial_g F(\bar{W},\bar{\Gamma},\bar{g}) = \bar{\Gamma} - \partial\Omega^*(\bar{g}) \ni 0.\end{aligned} \tag{21}$$

By the final inclusion and the convexity of Ω, it implies $\bar{g}\in\partial\Omega(\bar{\Gamma})$. Plugging this inclusion into the second inclusion yields $\alpha\nu^{-1}(\bar{\Gamma}-\bar{W})=0$. Together with the first equality imples

$$\nabla\bar{\mathcal{L}}(\bar{W},\bar{\Gamma})=0,\quad \nabla\widehat{\mathcal{L}}_n(\bar{W})=0.$$

This finishes the proof of this theorem.

A.4 Sufficient Descent Property along Lyapunov Function

Let $P_k := (W_k,\Gamma_k)$, and $Q_k := (P_k, g_{k-1}), k\in\mathbb{N}$. In the following, we present the sufficient descent property of Q_k along the Lyapunov function F.

Lemma. Suppose that $\widehat{\mathcal{L}}_n$ is continuously differentiable and $\nabla\widehat{\mathcal{L}}_n$ is Lipschitz continuous with a constant $Lip>0$. Let $\{Q_k\}$ be a sequence generated by SLBI with a finite initialization. If $0<\alpha<\frac{2}{\kappa(Lip+\nu^{-1})}$, then

$$F(Q_{k+1}) \le F(Q_k) - \rho\|Q_{k+1}-Q_k\|_2^2,$$

where $\rho := \frac{1}{\kappa} - \frac{\alpha(Lip+\nu^{-1})}{2}$.

Proof. By the optimality condition of (17a) and also the inclusion $p_k=[0,g_k]^T\in\partial R(P_k)$, there holds

$$\kappa(\alpha\nabla\bar{\mathcal{L}}(P_k)+p_{k+1}-p_k)+P_{k+1}-P_k=0,$$

which implies

$$-\langle\alpha\nabla\bar{\mathcal{L}}(P_k), P_{k+1}-P_k\rangle = \kappa^{-1}\|P_{k+1}-P_k\|_2^2 + D(\Gamma_{k+1},\Gamma_k) \tag{22}$$

where

$$D(\Gamma_{k+1},\Gamma_k) := \langle g_{k+1}-g_k, \Gamma_{k+1}-\Gamma_k\rangle.$$

Noting that $\bar{\mathcal{L}}(P)=\widehat{\mathcal{L}}_n(W)+\frac{1}{2\nu}\|W-\Gamma\|_2^2$ and by the Lipschitz continuity of $\nabla\widehat{\mathcal{L}}_n(W)$ with a constant $Lip>0$ implies $\nabla\bar{\mathcal{L}}$ is Lipschitz continuous with a constant $Lip+\nu^{-1}$. This implies

$$\bar{\mathcal{L}}(P_{k+1}) \le \bar{\mathcal{L}}(P_k) + \langle\nabla\bar{\mathcal{L}}(P_k), P_{k+1}-P_k\rangle + \frac{Lip+\nu^{-1}}{2}\|P_{k+1}-P_k\|_2^2.$$

Substituting the above inequality into (22) yields

$$\alpha\bar{\mathcal{L}}(P_{k+1}) + D(\Gamma_{k+1},\Gamma_k) + \rho\|P_{k+1}-P_k\|_2^2 \le \alpha\bar{\mathcal{L}}(P_k). \tag{23}$$

Adding some terms in both sides of the above inequality and after some reformulations implies

$$\begin{aligned}&\alpha\bar{\mathcal{L}}(P_{k+1}) + B_\Omega^{g_k}(\Gamma_{k+1},\Gamma_k) \\ &\le \alpha\bar{\mathcal{L}}(P_k) + B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1}) - \rho\|P_{k+1}-P_k\|_2^2 - \big(D(\Gamma_{k+1},\Gamma_k) + B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1}) - B_\Omega^{g_k}(\Gamma_{k+1},\Gamma_k)\big)\\ &= \alpha\bar{\mathcal{L}}(P_k) + B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1}) - \rho\|P_{k+1}-P_k\|_2^2 - B_\Omega^{g_{k+1}}(\Gamma_k,\Gamma_{k-1}) - B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1}),\end{aligned} \tag{24}$$

where the final equality holds for $D(\Gamma_{k+1},\Gamma_k) - B_\Omega^{g_k}(\Gamma_{k+1},\Gamma_k) = B_\Omega^{g_{k+1}}(\Gamma_k,\Gamma_{k-1})$. That is,

$$\begin{aligned}F(Q_{k+1}) &\le F(Q_k) - \rho\|P_{k+1}-P_k\|_2^2 - B_\Omega^{g_{k+1}}(\Gamma_k,\Gamma_{k-1}) - B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1}) &(25)\\ &\le F(Q_k) - \rho\|P_{k+1}-P_k\|_2^2, &(26)\end{aligned}$$

where the final inequality holds for $B_\Omega^{g_{k+1}}(\Gamma_k,\Gamma_{k-1})\ge 0$ and $B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1})\ge 0$. Thus, we finish the proof of this lemma. □

Based on Lemma A.4, we directly obtain the following lemma.

Lemma 1. *Suppose that assumptions of Lemma A.4 hold. Suppose further that Assumption 1 (b)-(d) hold. Then*

(i) both $\alpha\{\bar{\mathcal{L}}(P_k)\}$ and $\{F(Q_k)\}$ converge to the same finite value, and $\lim_{k\to\infty} B_\Omega^{g_k}(\Gamma_{k+1},\Gamma_k)=0$.

(ii) the sequence $\{(W_k,\Gamma_k,g_k)\}$ is bounded,

(iii) $\lim_{k\to\infty}\|P_{k+1}-P_k\|_2^2=0$ *and* $\lim_{k\to\infty}D(\Gamma_{k+1},\Gamma_k)=0$,

(iv) $\frac{1}{K}\sum_{k=0}^{K}\|P_{k+1}-P_k\|_2^2\to 0$ *at a rate of* $\mathcal{O}(1/K)$.

Proof. By (23), $\bar{\mathcal{L}}(P_k)$ is monotonically decreasing due to $D(\Gamma_{k+1},\Gamma_k)\geq 0$. Similarly, by (26), $F(Q^k)$ is also monotonically decreasing. By the lower boundedness assumption of $\widehat{\mathcal{L}}_n(W)$, both $\bar{\mathcal{L}}(P)$ and $F(Q)$ are lower bounded by their definitions, i.e., (4) and (11), respectively. Therefore, both $\{\bar{\mathcal{L}}(P_k)\}$ and $\{F(Q_k)\}$ converge, and it is obvious that $\lim_{k\to\infty}F(Q_k)\geq\lim_{k\to\infty}\alpha\bar{\mathcal{L}}(P_k)$. By (25),

$$B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1})\leq F(Q_k)-F(Q_{k+1}),\ k=1,\ldots.$$

By the convergence of $F(Q_k)$ and the nonegativeness of $B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1})$, there holds

$$\lim_{k\to\infty}B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1})=0.$$

By the definition of $F(Q_k)=\alpha\bar{\mathcal{L}}(P_k)+B_\Omega^{g_{k-1}}(\Gamma_k,\Gamma_{k-1})$ and the above equality, it yields

$$\lim_{k\to\infty}F(Q_k)=\lim_{k\to\infty}\alpha\bar{\mathcal{L}}(P_k).$$

Since $\widehat{\mathcal{L}}_n(W)$ has bounded level sets, then W_k is bounded. By the definition of $\bar{\mathcal{L}}(W,\Gamma)$ and the finiteness of $\bar{\mathcal{L}}(W_k,\Gamma_k)$, Γ_k is also bounded due to W_k is bounded. The boundedness of g_k is due to $g_k\in\partial\Omega(\Gamma_k)$, condition (d), and the boundedness of Γ_k.

By (26), summing up (26) over $k=0,1,\ldots,K$ yields

$$\sum_{k=0}^{K}\left(\rho\|P_{k+1}-P_k\|^2+D(\Gamma_{k+1},\Gamma_k)\right)<\alpha\bar{\mathcal{L}}(P_0)<\infty. \tag{27}$$

Letting $K\to\infty$ and noting that both $\|P_{k+1}-P_k\|^2$ and $D(\Gamma_{k+1},\Gamma_k)$ are nonnegative, thus

$$\lim_{k\to\infty}\|P_{k+1}-P_k\|^2=0,\quad \lim_{k\to\infty}D(\Gamma_{k+1},\Gamma_k)=0.$$

Again by (27),

$$\frac{1}{K}\sum_{k=0}^{K}\left(\rho\|P_{k+1}-P_k\|^2+D(\Gamma_{k+1},\Gamma_k)\right)<K^{-1}\alpha\bar{\mathcal{L}}(P_0),$$

which implies $\frac{1}{K}\sum_{k=0}^{K}\|P_{k+1}-P_k\|^2\to 0$ at a rate of $\mathcal{O}(1/K)$. □

A.5 Relative Error Property

In this subsection, we provide the bound of subgradient by the discrepancy of two successive iterates. By the definition of F (11),

$$H_{k+1}:=\begin{pmatrix}\alpha\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})\\ \alpha\nabla_\Gamma\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})+g_{k+1}-g_k\\ \Gamma_k-\Gamma_{k+1}\end{pmatrix}\in\partial F(Q_{k+1}),\ k\in\mathbb{N}. \tag{28}$$

Lemma. Under assumptions of Lemma 1, then

$$\|H_{k+1}\|\leq\rho_1\|Q_{k+1}-Q_k\|,\ \text{for } H_{k+1}\in\partial F(Q_{k+1}),\ k\in\mathbb{N},$$

where $\rho_1:=2\kappa^{-1}+1+\alpha(Lip+2\nu^{-1})$. Moreover, $\frac{1}{K}\sum_{k=1}^{K}\|H_k\|^2\to 0$ at a rate of $\mathcal{O}(1/K)$.

Proof. Note that

$$\begin{aligned}\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})=&(\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})-\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_k))\\&+(\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_k)-\nabla_W\bar{\mathcal{L}}(W_k,\Gamma_k))+\nabla_W\bar{\mathcal{L}}(W_k,\Gamma_k).\end{aligned} \tag{29}$$

By the definition of $\bar{\mathcal{L}}$ (see (4)),

$$\begin{aligned}\|\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})-\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_k)\|&=\nu^{-1}\|\Gamma_k-\Gamma_{k+1}\|,\\ \|\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_k)-\nabla_W\bar{\mathcal{L}}(W_k,\Gamma_k)\|&=\|(\nabla\widehat{\mathcal{L}}_n(W_{k+1})-\nabla\widehat{\mathcal{L}}_n(W_k))+\nu^{-1}(W_{k+1}-W_k)\|\\&\leq(Lip+\nu^{-1})\|W_{k+1}-W_k\|,\end{aligned}$$

Dataset		MNIST	Cifar-10	ImageNet-2012	
Models	Variants	LeNet	ResNet-20	AlexNet	ResNet-18
SGD	Naive	98.87	86.46	–/–	60.76/79.18
	l_1	98.52	67.60	46.49/65.45	51.49/72.45
	Mom	99.16	89.44	55.14/78.09	66.98/86.97
	Mom-W$d^\star$	**99.23**	**90.31**	56.55/79.09	69.76/89.18
	Nesterov	**99.23**	90.18	-/-	70.19/89.30
Adam	Naive	99.19	89.14	–/–	59.66/83.28
	Adabound	99.15	87.89	–/–	–/–
	Adagrad	99.02	88.17	–/–	–/–
	Amsgrad	99.14	88.68	–/–	–/–
	Radam	99.08	88.44	–/–	–/–
DessiLBI	Naive	99.02	89.26	55.06/77.69	65.26/86.57
	Mom	99.19	89.72	56.23/78.48	68.55/87.85
	Mom-Wd	99.20	89.95	**57.09/79.86**	**70.55/89.56**

Table 2. Top-1/Top-5 accuracy(%) on ImageNet-2012 and test accuracy on MNIST/Cifar-10. *: results from the official pytorch website. We use the official pytorch codes to run the competitors. All models are trained by 100 epochs. In this table, we run the experiment by ourselves except for SGD Mom-Wd on ImageNet which is reported in https://pytorch.org/docs/stable/torchvision/models.html.

where the last inequality holds for the Lipschitz continuity of $\nabla\widehat{\mathcal{L}}_n$ with a constant $Lip > 0$, and by (18a),

$$\|\nabla_W\bar{\mathcal{L}}(W_k,\Gamma_k)\| = (\kappa\alpha)^{-1}\|W_{k+1}-W_k\|.$$

Substituting the above (in)equalities into (29) yields

$$\|\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})\| \le \left[(\kappa\alpha)^{-1}+Lip+\nu^{-1}\right]\cdot\|W_{k+1}-W_k\|+\nu^{-1}\|\Gamma_{k+1}-\Gamma_k\|$$

Thus,

$$\|\alpha\nabla_W\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})\| \le \left[\kappa^{-1}+\alpha(Lip+\nu^{-1})\right]\cdot\|W_{k+1}-W_k\|+\alpha\nu^{-1}\|\Gamma_{k+1}-\Gamma_k\|. \tag{30}$$

By (18c), it yields

$$g_{k+1}-g_k = \kappa^{-1}(\Gamma_k-\Gamma_{k+1})-\alpha\nabla_\Gamma\bar{\mathcal{L}}(W_k,\Gamma_k).$$

Noting that $\nabla_\Gamma\bar{\mathcal{L}}(W_k,\Gamma_k)=\nu^{-1}(\Gamma_k-W_k)$, and after some simplifications yields

$$\begin{aligned}\|\alpha\nabla_\Gamma\bar{\mathcal{L}}(W_{k+1},\Gamma_{k+1})+g_{k+1}-g_k\| &= \|(\kappa^{-1}-\alpha\nu^{-1})\cdot(\Gamma_k-\Gamma_{k+1})+\alpha\nu^{-1}(W_k-W_{k+1})\| \\ &\le \alpha\nu^{-1}\|W_k-W_{k+1}\|+(\kappa^{-1}-\alpha\nu^{-1})\|\Gamma_k-\Gamma_{k+1}\|,\end{aligned} \tag{31}$$

where the last inequality holds for the triangle inequality and $\kappa^{-1}>\alpha\nu^{-1}$ by the assumption.

By (30), (31), and the definition of H_{k+1} (28), there holds

$$\begin{aligned}\|H_{k+1}\| &\le \left[\kappa^{-1}+\alpha(Lip+2\nu^{-1})\right]\cdot\|W_{k+1}-W_k\|+(\kappa^{-1}+1)\|\Gamma_{k+1}-\Gamma_k\| \\ &\le \left[2\kappa^{-1}+1+\alpha(Lip+2\nu^{-1})\right]\cdot\|P_{k+1}-P_k\| \\ &\le \left[2\kappa^{-1}+1+\alpha(Lip+2\nu^{-1})\right]\cdot\|Q_{k+1}-Q_k\|.\end{aligned} \tag{32}$$

By (32) and Lemma 1(iv), $\frac{1}{K}\sum_{k=1}^K\|H_k\|^2\to 0$ at a rate of $\mathcal{O}(1/K)$.

This finishes the proof of this lemma. □

B Supplementary Experiments

B.1 Ablation Study on Image Classification

Experimental Design. We compare different variants of SGD and Adam in the experiments. By default, the learning rate of competitors is set as 0.1 for SGD and its variant and 0.001 for Adam and its variants, and gradually decreased by 1/10 every 30 epochs. In particular, we have,

SGD: (1) Naive SGD: the standard SGD with batch input. (2) SGD with l_1 penalty (Lasso). The l_1 norm is applied to penalize the weights of SGD by encouraging the sparsity of learned model, with the regularization parameter of the l_1 penalty

term being set as $1e^{-3}$ (3) SGD with momentum (Mom): we utilize momentum 0.9 in SGD. (4) SGD with momentum and weight decay (Mom-Wd): we set the momentum 0.9 and the standard l_2 weight decay with the coefficient weight $1e^{-4}$. (5) SGD with Nesterov (Nesterov): the SGD uses nesterov momentum 0.9.

Adam: (1) Naive Adam: it refers to the standard version of Adam. We report the results of several recent variants of Adam, including (2) Adabound, (3) Adagrad, (4) Amsgrad, and (5) Radam.

The results of image classification are shown in Tab. 2 . It shows the experimental results on ImageNet-2012, Cifar-10, and MNIST of some classical networks -- LeNet, AlexNet and ResNet. Our DessiLBI variants may achieve comparable or even better performance than SGD variants in 100 epochs, indicating the efficacy in learning dense, over-parameterized models. The visualization of learned ResNet-18 on ImageNet-2012 is given in Fig. 6.

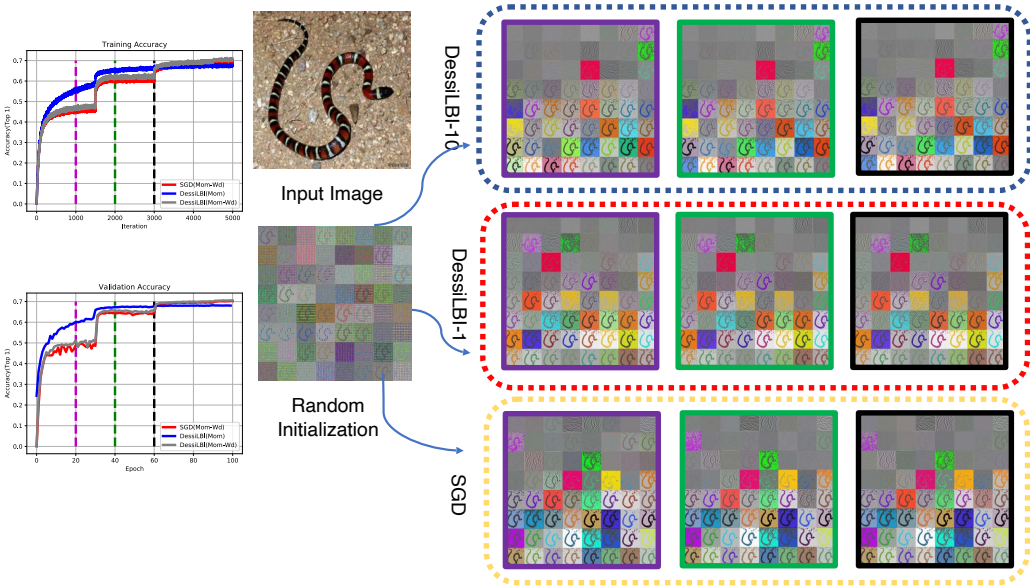


Figure 6. Visualization of the first convolutional layer filters of ResNet-18 trained on ImageNet-2012. Given the input image and initial weights visualized in the middle, filter response gradients at 20 (purple), 40 (green), and 60 (black) epochs are visualized by (Springenberg et al., 2014).

B.2 Ablation Study of VGG16 and ResNet56 on Cifar10

To further study the influence of hyperparameters, we record performance of W_t for each epoch t with different combinations of hyperparameters. The experiments is conducted 5 times each, we show the mean in the table, the standard error can be found in the corresponding figure. We perform experiments on Cifar10 and two commonly used network VGG16 and ResNet56.

On κ , we keep $\nu = 100$ and try $\kappa = 1, 2, 5, 10$, the validation curves of models W_t are shown in Fig. 7 and Table 3 summarizes the mean accuracies. Table 4 summarizes best validation accuracies achieved at some epochs, together with their sparsity rates. These results show that larger kappa leads to slightly lower validation accuracies, where the numerical results are shown in Table 3 . We can find that $\kappa = 1$ achieves the best test accuracy.

On ν , we keep $\kappa = 1$ and try $\nu = 10, 20, 50, 100, 200, 500, 1000, 2000$ the validation curve and mean accuracies are show in Fig. 7 and Table 5. Table 6 summarizes best validation accuracies achieved at some epochs, together with their sparsity rates. By carefully tuning ν we can achieve similar or even better results compared to SGD. Different from κ, ν has less effect on the generalization performance. By tuning it carefully, we can even get a sparse model with slightly better performance than SGD trained model.

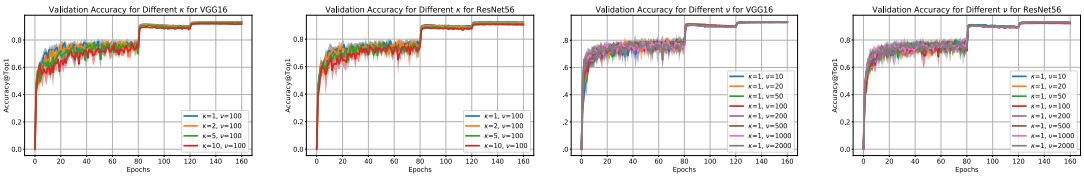


Figure 7. Validation curves of dense models W_t for different κ and ν. For DessiLBI we find that the model accuracy is robust to the hyperparameters both in terms of convergence rate and generalization ability. Here validation accuracy means the accuracy on test set of Cifar10. The first one is the result for VGG16 ablation study on κ, the second one is the result for ResNet56 ablation study on κ, the third one is the result for VGG16 ablation study on ν and the forth one is the result for ResNet56 ablation study on ν.

Type	Model	$\kappa = 1$	$\kappa = 2$	$\kappa = 5$	$\kappa = 10$	SGD
Full	Vgg16	93.46	93.27	92.77	92.03	93.57
	ResNet56	92.71	92.18	91.50	90.92	93.08
Sparse	Vgg16	93.31	93.00	92.36	76.25	-
	ResNet56	92.37	91.85	89.48	87.02	-

Table 3. This table shows results for different κ, the results are all the best test accuracy. Here we test two widely-used models: VGG16 and ResNet56 on Cifar10. For results in this table, we keep $\nu = 100$. Full means that we use the trained model weights directly, Sparse means the model weights are combined with mask generated by Γ support. Sparse result has no finetuning process, the result is comparable to its Full counterpart. For this experiment, we propose that $\kappa = 1$ is a good choice. For all the model, we train for 160 epochs with initial learning rate (lr) of 0. 1 and decrease by 0.1 at epoch 80 and 120.

Model		Ep20		Ep40		Ep80		Ep160	
Vgg16	Term	Sparsity	Acc	Spasity	Acc	Spasity	Acc	Spasity	Acc
	$\kappa = 1$	96.62	71.51	96.62	76.92	96.63	77.48	96.63	93.31
	$\kappa = 2$	51.86	72.98	71.99	73.64	75.69	74.54	75.72	93.00
	$\kappa = 5$	8.19	10.00	17.64	34.25	29.76	69.92	30.03	92.36
	$\kappa = 10$	0.85	10.00	6.62	10.00	12.95	38.38	13.26	76.25
ResNet56	Term	Sparsity	Acc	Spasity	Acc	Spasity	Acc	Spasity	Acc
	$\kappa = 1$	96.79	73.50	96.87	75.27	96.69	77.47	99.68	92.37
	$\kappa = 2$	76.21	72.85	81.41	74.72	84.17	75.64	84.30	91.85
	$\kappa = 5$	36.58	60.43	53.07	76.00	57.48	75.67	57.74	89.48
	$\kappa = 10$	3.12	10.20	29.43	53.36	41.18	74.56	41.14	87.02

Table 4. Sparsity rate and validation accuracy for different κ at different epochs. Here we pick the test accuracy for specific epoch. In this experiment, we keep $\nu = 100$. We pick epoch 20, 40, 80 and 160 to show the growth of sparsity and sparse model accuracy. Here Sparsity is defined in Sec. 5, and Acc means the test accuracy for sparse model. A sparse model is a model at designated epoch t combined with the mask as the support of Γ_t.

Type	Model	$\nu = 10$	$\nu = 20$	$\nu = 50$	$\nu = 100$	$\nu = 200$	$\nu = 500$	$\nu = 1000$	$\nu = 2000$	SGD
Full	Vgg16	93.66	93.59	93.57	93.39	93.38	93.35	93.43	93.46	93.57
	ResNet56	93.12	92.68	92.78	92.45	92.95	93.11	93.16	93.31	93.08
Sparse	Vgg16	93.39	93.42	93.39	93.23	93.21	93.01	92.68	10	-
	ResNet56	92.81	92.19	92.40	92.10	92.68	92.81	92.84	88.96	-

Table 5. Results for different ν, the results are all the best test accuracy. Here we test two widely-used model : VGG16 and ResNet56 on Cifar10. For results in this table, we keep $\kappa = 1$. Full means that we use the trained model weights directly, Sparse means the model weights are combined with mask generated by Γ support. Sparse result has no finetuning process, the result is comparable to its Full counterpart. For all the model, we train for 160 epochs with initial learning rate (lr) of 0.1 and decrease by 0.1 at epoch 80 and 120.

Model		Ep20		Ep40		Ep80		Ep160	
	Term	Sparsity	Acc	Spasity	Acc	Spasity	Acc	Spasity	Acc
Vgg16	$\nu = 10$	96.64	71.07	96.64	77.70	96.65	79.46	96.65	93.34
	$\nu = 20$	96.64	69.11	96.64	77.63	96.65	77.08	96.65	93.42
	$\nu = 50$	96.64	74.91	96.65	74.21	96.65	79.15	96.65	93.38
	$\nu = 100$	96.64	74.82	96.64	73.22	96.64	78.09	96.64	93.23
	$\nu = 200$	91.69	73.67	94.06	74.67	94.15	75.20	94.15	93.21
	$\nu = 500$	18.20	10.00	59.94	67.88	82.03	78.69	82.32	93.01
	$\nu = 1000$	6.43	10.00	17.88	10.00	49.75	61.31	51.21	92.68
	$\nu = 2000$	0.22	10.00	6.89	10.00	18.15	10.00	19.00	10.00
	Term	Sparsity	Acc	Spasity	Acc	Spasity	Acc	Spasity	Acc
ResNet56	$\nu = 10$	99.97	73.37	99.95	71.64	99.74	76.46	99.74	92.81
	$\nu = 20$	99.97	72.58	99.84	74.16	99.69	72.37	99.72	92.19
	$\nu = 50$	99.96	70.72	99.89	73.96	99.79	74.93	99.77	92.40
	$\nu = 100$	96.31	73.63	96.63	75.79	96.55	72.94	96.57	92.10
	$\nu = 200$	91.98	75.30	94.38	72.13	94.87	73.75	94.88	92.68
	$\nu = 500$	74.44	65.58	90.00	74.12	92.96	71.91	92.99	92.81
	$\nu = 1000$	24.32	10.85	75.68	70.23	88.56	79.67	88.80	92.48
	$\nu = 2000$	0.65	10.00	26.66	13.30	74.98	70.38	75.92	88.95

Table 6. Sparsity rate and validation accuracy for different ν at different epochs. Here we pick the test accuracy for specific epoch. In this experiment, we keep $\kappa = 1$. We pick epoch 20, 40, 80 and 160 to show the growth of sparsity and sparse model accuracy. Here Sparsity is defined in Sec. 5 as the percentage of nonzero parameters, and Acc means the test accuracy for sparse model. A sparse model is a model at designated epoch t combined with mask as the support of Γ_t.

optimizer	SGD	DessiLBI	Adam
Mean Batch Time	0.0197	0.0221	0.0210
GPU Memory	1161MB	1459MB	1267MB

Table 7. Computational and Memory Costs.

C Computational Cost of DessiLBI

We further compare the computational cost of different optimizers: SGD (Mom), DessiLBI (Mom) and Adam (Naive). We test each optimizer on one GPU, and all the experiments are done on one GTX2080. For computational cost, we judge them from two aspects : GPU memory usage and time needed for one batch. The batch size here is 64, experiment is performed on VGG-16 as shown in Table 7.

D Fine-tuning of sparse subnetworks

We design the experiment on MNIST, inspired by (Frankle & Carbin, 2019). Here, we explore the subnet obtained by Γ_T after $T = 100$ epochs of training. As in (Frankle et al., 2019), we adopt the "rewind" trick: re-loading the subnet mask of Γ_{100} at different epochs, followed by fine-tuning. In particular, along the training paths, we reload the subnet models at Epoch 0, Epoch 30, 60, 90, and 100, and further fine-tune these models by DessiLBI (Mom-Wd). All the models use the same initialization and hence the subnet model at Epoch 0 gives the retraining with the same random initialization as proposed to find winning tickets of lottery in (Frankle & Carbin, 2019). We will denote the rewinded fine-tuned model at epoch 0 as (Lottery), and those at epoch 30, 60, 90, and 100, as F-epoch30, F-epoch60, F-epoch90, and F-epoch100, respectively. Three networks are studied here – LeNet-3, Conv-2, and Conv-4. LeNet-3 removes one convolutional layer of LeNet-5; and it is thus less over-parameterized than the other two networks. Conv-2 and Conv-4, as the scaled-down variants

Layer	FC1	FC2	FC3
Sparsity	0.049	0.087	0.398
Number of Weights	235200	30000	1000

Table 8. This table shows the sparsity for every layer of Lenet-3. Here sparsity is defined in Sec. 5, number of weights denotes the total number of parameters in the designated layer. It is interesting that the Γ tends to put lower sparsity on layer with more parameters.

Layer	Conv1	Conv2	FC1	FC2	FC3
Sparsity	0.9375	1	0.0067	0.0284	0.1551
Number of Weights	576	36864	3211264	65536	2560

Table 9. This table shows the sparsity for every layer of Conv-2. Here sparsity is defined in Sec. 5, number of weights denotes the total number of parameters in the designated layer. The sparsity is more significant in fully connected (FC) layers than convolutional layers.

Layer	Conv1	Conv2	Conv3	Conv4	FC1	FC2	FC3
Sparsity	0.921875	1	1	1	0.0040	0.0094	0.1004
Number of Weights	576	36864	73728	147456	1605632	65536	2560

Table 10. This table shows the sparsity for every layer of Conv-4. Here sparsity is defined in Sec. 5, number of weights denotes the total number of parameters in the designated layer. Most of the convolutional layers are kept while the FC layers are very sparse.

of VGG family as done in (Frankle & Carbin, 2019), have two and four fully-connected layers, respectively, followed by max-pooling after every two convolutional layer.

The whole sparsity for Lenet-3 is 0.055, Conv-2 is 0.0185, and Conv-4 is 0.1378. Detailed sparsity for every layer of the model is shown in Table 8, 9, 10. We find that fc-layers are sparser than conv-layers.

We compare DessiLBI variants to the SGD (Mom-Wd) and SGD (Lottery) (Frankle & Carbin, 2019) in the same structural sparsity and the results are shown in Fig. 8. In this exploratory experiment, one can see that for overparameterized networks – Conv-2 and Conv-4, fine-tuned rewinding subnets – F-epoch30, F-epoch60, F-epoch90, and F-epoch100, can produce *better* results than the full models; while for the less over-parameterized model LeNet-3, fine-tuned subnets may achieve less yet still comparable performance to the dense models and remarkably better than the retrained sparse subnets from beginning (i.e. DessiLBI/SGD (Lottery)). These phenomena suggest that the subnet architecture disclosed by structural sparsity parameter Γ_T is valuable, for fine-tuning sparse models with comparable or even better performance than the dense models of W_T.

E Retraining of sparse subnets found by DessiLBI (Lottery)

Here we provide more details on the experiments in Fig. 5. Table 11 gives the details on hyper-parameter setting. Moreover, Figure 9 provides the sparsity variations during DessiLBI training in Fig. 5.

Network	Penalty	Optimizer	α	ν	κ	λ	Momentum	Nesterov
VGG-16	Group Lasso	DessiLBI	0.1	100	1	0.1	0.9	Yes
ResNet-56	Group Lasso	DessiLBI	0.1	100	1	0.05	0.9	Yes
VGG-16(Lasso)	Lasso	DessiLBI	0.1	500	1	0.05	0.9	Yes
ResNet-50(Lasso)	Lasso	DessiLBI	0.1	200	1	0.03	0.9	Yes

Table 11. Hyperparameter setting for the experiments in Figure 5.

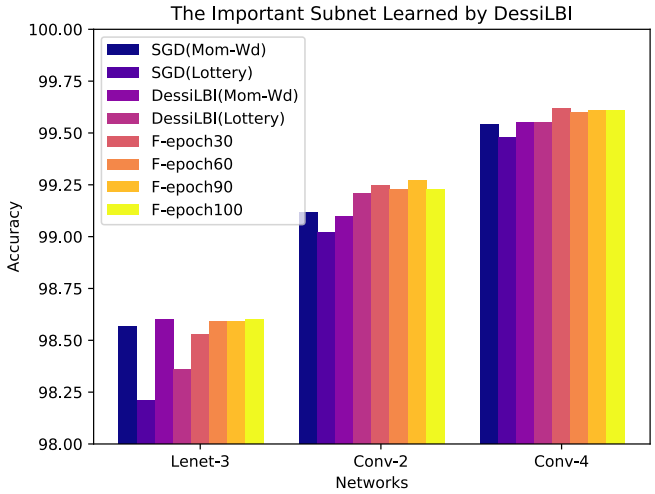
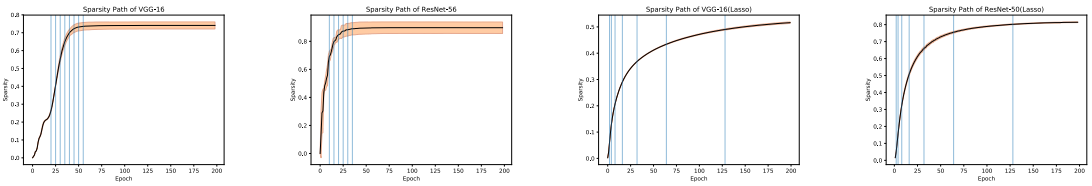


Figure 8. Fine-tuning of sparse subnets learned by DessiLBI may achieve comparable or better performance than dense models. F-epoch$k$ indicates the fine-tuned model comes from the Epoch k. DessiLBI (Lottery) and SGD (Lottery) use the same sparsity rate for each layer and the same initialization for retrain.



(a) VGG-16 (b) ResNet-56 (c) VGG-16 (Lasso) (d) ResNet-50 (Lasso)

Figure 9. Sparsity changing during training process of DessiLBI (Lottery) for VGG and ResNets (corresponding to Fig. 5). We calculate the sparsity in every epoch and repeat five times. The black curve represents the mean of the sparsity and shaded area shows the standard deviation of sparsity. The vertical blue line shows the epochs that we choose to early stop. We choose the log-scale epochs for achieve larger range of sparsity.