A Survey of Generalization in Randomized Optimization Algorithms

Jiahuan Wang, Xiaoge Deng, Ziqing Wen, Ping Luo, Dongsheng Li, Tao Sun* and Xinwang Liu *Senior Member, IEEE*

***Abstract*—Randomized optimization algorithms are central to modern machine learning, especially in large-scale and overparameterized model training. Their generalization behavior cannot be fully understood by classical capacity control over a fixed hypothesis space, because randomness enters directly into the update rule, shapes the optimization trajectory, and influences the final output selected by training. This survey reviews theoretical approaches for analyzing the generalization of randomized optimization algorithms from an algorithm-centered viewpoint. We first clarify the scope of randomized optimization considered in this survey and introduce the common mathematical objects used throughout the paper. We then organize the literature around four representative perspectives: stability-based analysis, which tracks the propagation of sample perturbations along the training trajectory; information-theoretic analysis, which controls the dependence between the training data and the algorithmic output; PAC-Bayesian analysis, which characterizes the prior-posterior complexity of randomized predictors; and algorithm-dependent complexity analysis, which localizes capacity control to the effective classes, trajectories, compressed representations, or geometric supports induced by training. Across these perspectives, we emphasize stochastic gradient methods and their variants, including noisy, distributed, and structured stochastic optimization settings. The survey aims to synthesize a scattered literature, clarify the connections among different theoretical tools, and highlight open challenges in developing a unified theory of generalization for randomized optimization.**

***Index Terms*—Randomized optimization algorithms, generalization analysis, algorithmic stability, information-theoretic generalization, PAC-Bayes, algorithm-dependent complexity, stochastic gradient methods.**

I. Introduction

GENERALIZATION is a central problem in statistical learning theory: a learning procedure is useful only if its performance on a finite training sample transfers to unseen data. Classical theory formalizes this problem through probabilistic and complexity-based viewpoints. PAC learning describes learnability in terms of accuracy, confidence, and sample size [1], [2]; VC dimension and Rademacher complexity control the deviation between empirical and population risks through the capacity of a hypothesis class [3]–[5]; and algorithmic stability explains generalization through the sensitivity of a learning algorithm to small perturbations of the training data [6], [7]. These ideas provide the basic language of generalization analysis, but they were largely developed around either static hypothesis classes or abstract learning rules.

Modern machine learning has shifted attention from static classes to training procedures. Large-scale models are typically trained by stochastic and iterative algorithms, such as stochastic gradient descent (SGD), mini-batch methods, random reshuffling, stochastic coordinate methods, noisy gradient methods, and distributed stochastic optimization. In these methods, randomness enters the update rule, affects the whole training trajectory, and influences which solution is eventually selected. Generalization is therefore not only a question of whether a hypothesis class is too large; it is also a question of how a concrete randomized optimization process transforms finite data, stochastic updates, and algorithmic design choices into a final predictor. This dynamic viewpoint is especially important in overparameterized learning, where classical capacity measures alone often fail to explain the observed generalization behavior [8]–[10].

This survey focuses on the generalization analysis of *randomized optimization algorithms*. We use this term in a relatively narrow sense: randomness should enter directly into the parameter update, the training trajectory, or the output distribution of the optimizer. This includes SGD-type algorithms, stochastic momentum and adaptive methods driven by mini-batch gradients, randomized coordinate methods, random reshuffling, Langevin-type algorithms, and distributed stochastic methods. Other forms of randomness in learning, such as random features, random projections, Dropout, stochastic depth, and data augmentation, are important but are not the main focus of this survey, because their primary randomness lies in representation, architecture, or data transformation rather than in the optimization dynamics themselves.

Several theoretical routes have been developed to study this problem. Stability-based analysis controls how replacing one training example changes the output or trajectory of the algorithm. PAC-Bayesian analysis views training as producing a data-dependent randomized predictor and controls its complexity through a prior-posterior divergence. Information-theoretic analysis measures how much information about the training sample is retained by the output model. Algorithm-dependent complexity analysis keeps the spirit of uniform convergence, but replaces the global hypothesis class by the effective class, trajectory set, compressed representation, or geometric support actually explored by the algorithm. These perspectives are technically different, but they address the

Corresponding author: Tao Sun.

Jiahuan Wang, Ziqing Wen, Ping Luo, Dongsheng Li, Tao Sun, Xinwang Liu are with the National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Changsha, 410073, China (e-mail: wangjiahuan@nudt.edu.cn; zqwen@nudt.edu.cn; luoping@nudt.edu.cn; dsli@nudt.edu.cn; suntao.saltfish@outlook.com; xinwangliu@nudt.edu.cn.)

Xiaoge Deng is the Intelligent Game and Decision Lab, Beijing, China (e-mail: dengxg@ustc.edu;)

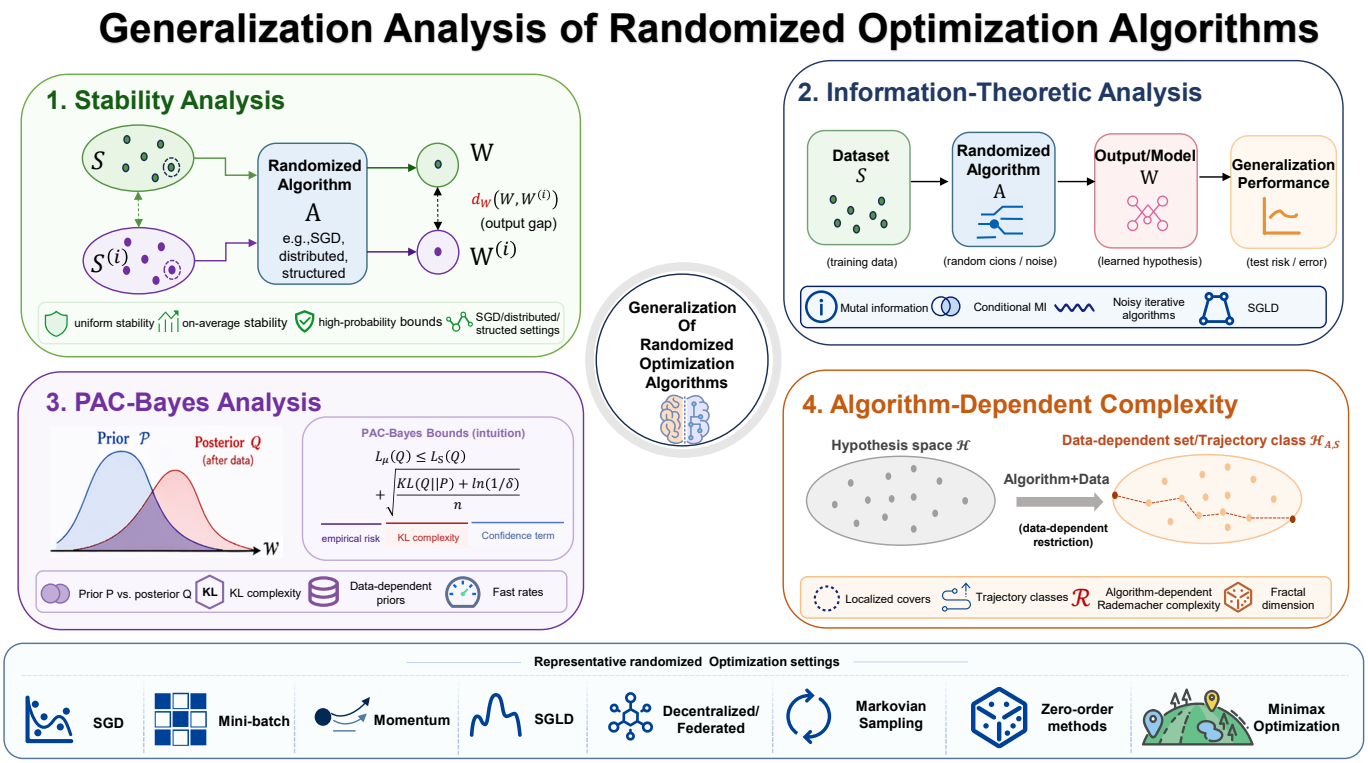

Fig. 1. Unified perspectives on the generalization analysis of randomized optimization algorithms.

same underlying question: which mathematical object associated with a randomized optimizer is sufficient to control its generalization gap? Fig. 1 provides a visual overview of these four routes, highlighting the distinct objects through which they control generalization.

The goal of this survey is to provide a systematic account of these four perspectives and to clarify their roles in the study of randomized optimization. Rather than treating the literature as a collection of isolated bounds, we organize it around the controlled object of each theory: perturbation propagation for stability, data-output dependence for information-theoretic methods, posterior complexity for PAC-Bayes, and effective algorithm-induced classes for algorithm-dependent uniform convergence. This organization highlights both the common structure and the differences among existing results, and it helps explain how generalization theory has gradually moved from static capacity control toward algorithm-centered analysis.

The remainder of the paper is organized as follows. Section II introduces the basic notation, problem setup, scope of randomized optimization algorithms, and the four mathematical objects that will be used to organize the survey. Sections III–VI review stability-based analysis, information-theoretic analysis, PAC-Bayesian analysis, and algorithm-dependent complexity analysis, respectively. Finally, Section VII discusses broader connections, limitations, and future directions.

II. Preliminaries

This section establishes the common mathematical language used throughout the survey. Its purpose is not to provide a textbook review of PAC learning, VC theory, Rademacher complexity, stability, PAC-Bayes, or information theory. Instead, we introduce the learning setup, define randomized optimizers as stochastic learning procedures, distinguish the main types of generalization guarantees, and identify the four analytical objects that will organize the later sections. For ease of reference, the core notation used throughout the survey is summarized in Table I.

A. Learning Setup

Let $\mathcal{Z}$ be a sample space and let μ be an unknown distribution over $\mathcal{Z}$. A training sample is denoted by $\mathcal{S} = (Z_1, \ldots, Z_n) \sim \mu^{\otimes n}$, unless otherwise specified. For a parameter or hypothesis $\mathbf{w} \in \mathcal{W}$ and a loss function $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$, the population and empirical risks are defined as

$$L_\mu(\mathbf{w}) = \mathbb{E}_{Z\sim\mu}[\ell(\mathbf{w}; Z)], \quad L_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}; Z_i). \quad (1)$$

The goal of learning is to output a parameter W with small population risk. Since μ is unknown, training algorithms usually minimize or approximately minimize the empirical risk, possibly with regularization, noise injection, or algorithmic constraints.

A learning algorithm is written as a possibly randomized map $W = A(\mathcal{S}, \xi)$, where ξ denotes all algorithmic randomness. Equivalently, A can be viewed as a stochastic

kernel $P_{W|\mathcal{S}}$ from the sample space $\mathcal{Z}^n$ to the parameter space $\mathcal{W}$. This stochastic-kernel viewpoint is useful because it provides a common language for stability analysis, PAC-Bayesian analysis, and information-theoretic analysis: the same randomized optimizer can be studied through coupled outputs, a training-induced posterior, or a data-output information measure. Throughout the paper, we use $\mathcal{S}^{(i)}$ to denote a neighboring dataset obtained by replacing the i-th example in $\mathcal{S}$ with an independent copy Z_i'. In some stability arguments, we also use $\mathcal{S}^{\setminus i}$ to denote the sample with Z_i removed. These neighboring samples allow one to formalize how sensitive an algorithm is to a small perturbation of the training data.

B. Randomized Optimization Algorithms

The main object of this survey is not an arbitrary randomized predictor, but a predictor generated by a randomized optimization process. A generic iterative randomized optimizer can be written as

$$\mathbf{w}_{t+1} = \Phi_t(\mathbf{w}_t, \mathcal{S}, \xi_t), \qquad t = 0, \dots, T-1, \tag{2}$$

where ξ_t denotes the randomness used at iteration t. The final output W may be the last iterate $\mathbf{w}_T$, an averaged iterate, a randomly selected iterate, or a randomized perturbation of the trajectory. This formulation includes a broad range of algorithms.

For example, mini-batch SGD has the update

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \frac{1}{|B_t|} \sum_{i \in B_t} \nabla \ell(\mathbf{w}_t; Z_i), \tag{3}$$

where $B_t \subseteq \{1, \dots, n\}$ is a random mini-batch and η_t is the stepsize. Random reshuffling replaces independent mini-batch sampling with sampling without replacement within each epoch. Noisy gradient methods, including Langevin-type algorithms, further inject explicit noise:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t g_t + \sigma_t \zeta_t, \tag{4}$$

where g_t is a stochastic gradient estimate and ζ_t is an exogenous noise variable. In decentralized stochastic optimization, m agents maintain local parameters and combine stochastic gradients with communication:

$$\mathbf{w}_{k,t+1} = \sum_{j=1}^{m} a_{kj} \mathbf{w}_{j,t} - \eta_t g_{k,t}, \qquad k = 1, \dots, m, \tag{5}$$

where $\mathbf{A} = (a_{kj})$ is a mixing matrix and $g_{k,t}$ is the k-th agent's stochastic gradient estimate. These algorithms differ in implementation, but they share the same theoretical feature: the learned model is generated by a random trajectory rather than selected by a deterministic empirical-risk minimization oracle.

C. Generalization Gap and Types of Guarantees

For a randomized optimizer $W = A(\mathcal{S}, \xi)$, the generalization gap is defined as

$$\mathrm{gen}(A, \mathcal{S}) = L_\mu(W) - L_\mathcal{S}(W). \tag{6}$$

TABLE I
CORE NOTATION USED THROUGHOUT THE SURVEY.

Symbol	Meaning
$\mathcal{Z}, \mu$	Sample space; data distribution
$\mathcal{S}$	Training sample
$\mathcal{S}^{(i)}$	Neighboring sample
$A(\mathcal{S}, \xi)$	Randomized optimizer
ξ, ξ_t	Algorithmic randomness
$P_{W\|\mathcal{S}}$	Algorithm-induced kernel
W	Final random output
w_t, W_t	Realized iterate; random iterate
$w_t^{(i)}, W_t^{(i)}$	Coupled iterate on $\mathcal{S}^{(i)}$
$L_\mu(w)$	Population risk
$L_\mathcal{S}(w)$	Empirical risk
$\mathrm{gen}(A, \mathcal{S})$	Generalization gap
Δ_t	Trajectory perturbation
ϵ_{stab}	Stability coefficient
P, Q	Prior; posterior
$\mathrm{KL}(Q\|P)$	Prior-posterior divergence
$I(\mathcal{S}; W)$	Data-output mutual information
$\mathcal{H}_{A,\mathcal{S}}$	Algorithm-induced class
$\mathcal{T}_{A,\mathcal{S}}$	Trajectory-induced set
$\widehat{\mathfrak{R}}_\mathcal{S}(\cdot)$	Empirical Rademacher complexity

Depending on the theory, this quantity can be controlled in different senses.

The first form is an expected generalization bound:

$$\left|\mathbb{E}_{\mathcal{S},\xi}\left[\mathrm{gen}(A, \mathcal{S})\right]\right| \le \varepsilon(n, T, \eta, \sigma, \dots),$$

where the right-hand side may depend on the sample size, training time, stepsizes, noise level, batch size, dimension, or network topology. Many stability and information-theoretic results first appear in this form.

The second form is a high-probability bound:

$$\mathbb{P}\left(L_\mu(W) - L_\mathcal{S}(W) \le \varepsilon(n, \delta)\right) \ge 1 - \delta.$$

Such bounds provide stronger statements about typical training samples and random runs. They are often technically more difficult, especially for iterative randomized algorithms, because one must control not only the average behavior of the algorithm but also the concentration of its data-dependent trajectory.

The third form concerns randomized predictors or posterior distributions. If the optimizer induces a distribution Q over predictors, one may study the averaged risks

$$L_\mu(Q) = \mathbb{E}_{W\sim Q}[L_\mu(W)], \; L_\mathcal{S}(Q_\mathcal{S}) = \mathbb{E}_{W\sim Q}[L_\mathcal{S}(W)]. \tag{7}$$

PAC-Bayesian bounds typically control the deviation between these two quantities through a divergence between Q and a prior distribution P. This viewpoint is particularly natural when the learning algorithm itself is randomized or when one studies perturbations around a learned solution.

These three types of guarantees should not be conflated. Expected bounds reveal the average effect of algorithmic randomness; high-probability bounds describe typical outcomes; and posterior-based bounds characterize the complexity of a randomized predictor distribution. Many modern results combine elements of more than one type.

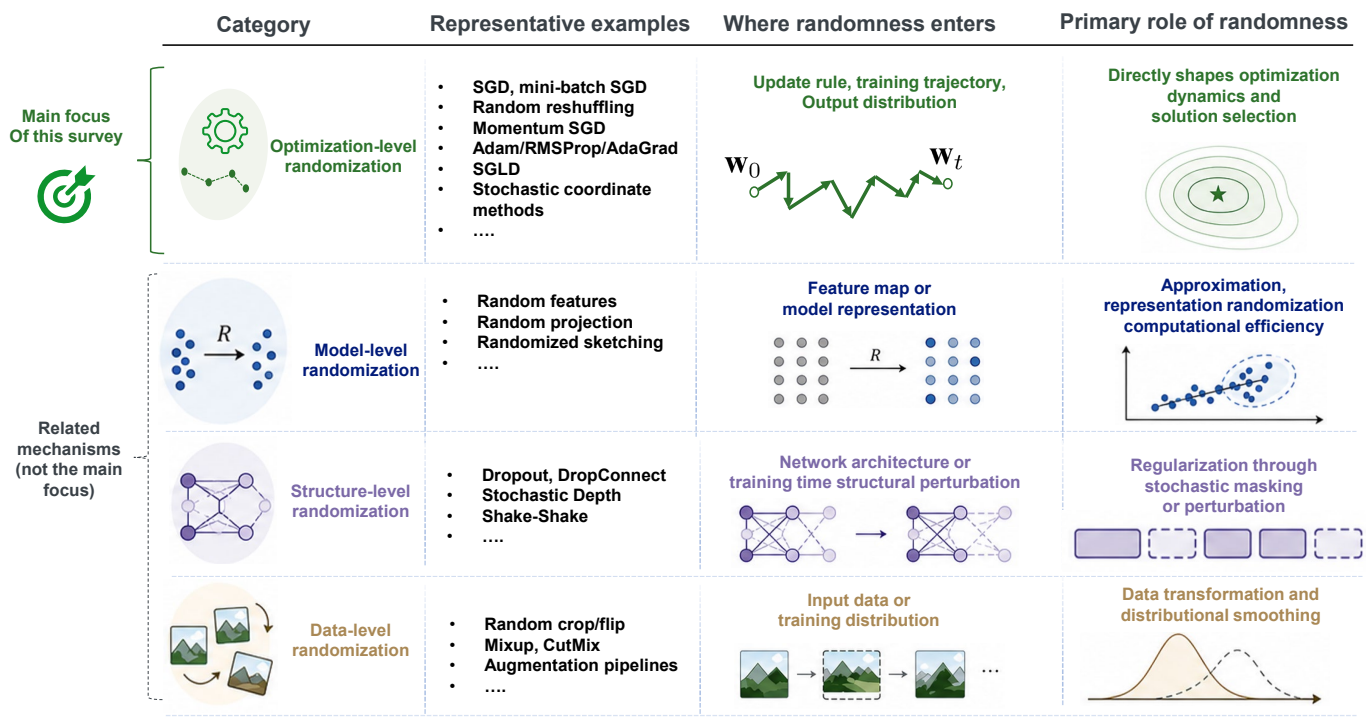

Fig. 2. Unified perspectives on the generalization analysis of randomized optimization algorithms.

D. Scope of Randomization

Randomness in learning can enter at several levels. To keep the survey focused, we distinguish optimization-level randomization from model-level, structure-level, and data-level randomization. The former is the main object of this survey because it directly governs the evolution of the training trajectory in parameter space. It is therefore the mechanism through which sample perturbations, noise structure, local geometry, and algorithmic design choices are transmitted to the final model. Figure 2 summarizes this distinction and clarifies which types of randomness are treated as the main focus of this survey.

This distinction does not imply that other random mechanisms are unimportant. In practice, stochastic optimization often interacts with data augmentation, architectural perturbations, explicit regularization, and model randomization. However, treating all sources of randomness on equal footing would make the survey too broad. We therefore focus on randomness that directly shapes the optimization dynamics, while discussing other mechanisms only when they clarify the generalization behavior of randomized optimization algorithms. Within this scope, the remainder of the survey is organized around four main theoretical tools. Table II lists representative works for each tool and summarizes the role that each perspective plays in our organization.

III. Stability-based analysis

Algorithmic stability studies generalization through the sensitivity of the learning algorithm to sample-level perturbations. Instead of controlling the size of the entire hypothesis space, it asks whether replacing one example in the training set can significantly change the output, the prediction loss, or the optimization trajectory of the algorithm. This idea can be traced back to early perturbation analyses of learning rules and error estimates [58]–[61], and was later developed into a systematic generalization framework by Bousquet and Elisseeff [6], [62]. For randomized optimization algorithms, stability is especially natural: the output is generated through a long sequence of local random updates, so the key technical question becomes how a one-sample perturbation is amplified, damped, or averaged along the training trajectory [7], [14], [15], [17], [21], [63]–[71].

A. From Sample Perturbation to Generalization

Let $W = A(S,\xi)$ and $W^{(i)} = A(S^{(i)},\xi)$ denote two outputs generated with coupled algorithmic randomness. A basic stability coefficient measures

$$\epsilon_{\mathrm{stab}} = \sup_{S \simeq S^{(i)}} \sup_{z \in \mathcal{Z}} \left| \mathbb{E}_{\xi}\left[\ell(A(S,\xi);z) - \ell(A(S^{(i)},\xi);z)\right] \right|.$$

If ϵ_{stab} is small, then the algorithm cannot fit an individual training example in a way that strongly changes its behavior on a fresh test point. The standard stability-to-generalization implication [6], [11] is therefore

$$\left| \mathbb{E}_{S,\xi}\left[L_{\mu}(A(S,\xi)) - L_S(A(S,\xi))\right] \right| \le \epsilon_{\mathrm{stab}},$$

with high-probability counterparts available under boundedness assumptions. For example, if $0 \le \ell(w;z) \le B$ and A is ϵ-uniformly stable, Bousquet and Elisseeff [6] showed that, with probability at least $1-\delta$,

$$L_{\mu}(A(S)) - L_S(A(S)) \le 2\epsilon + (4n\epsilon + B)\sqrt{\frac{\log(1/\delta)}{2n}}.$$

Thus, the role of stability is not merely qualitative. It converts a perturbation estimate for the algorithm into a quantitative generalization bound.

For iterative randomized optimizers, it is often more convenient to control parameter perturbations rather than loss perturbations directly. This leads to on-average model stability [7], [13]. If

$$\mathbb{E}_{S,S',\xi}\left[\frac{1}{n}\sum_{i=1}^{n}\|A(S,\xi)-A(S^{(i)},\xi)\|\right]\le\epsilon,$$

then, for an L-Lipschitz loss, Lei and Ying [13] proved the expected generalization bound

$$\left|\mathbb{E}_{S,\xi}\left[L_\mu(A(S,\xi))-L_S(A(S,\xi))\right]\right|\le L\epsilon.$$

For smooth losses, an ℓ_2 version further relates generalization to squared model perturbations and empirical risk along the trajectory. This shift from loss stability to model stability is important for stochastic optimization because it allows the proof to follow the evolution of coupled iterates.

B. Stability Analysis of SGD: Main Developments

A central turning point in stability-based analysis of randomized optimization is the work of Hardt et al. [11], which showed that the generalization behavior of SGD can be studied directly from the sensitivity of its optimization trajectory. This result is important not merely because it provides bounds for SGD, but because it changes the object of analysis: instead of asking how large the hypothesis class is, one asks how a one-sample perturbation propagates through the stochastic update rule. In this sense, SGD stability gives one of the clearest examples of how optimization parameters, such as stepsizes and training time, become statistical regularizers.

For empirical risk minimization, SGD takes the form

$$\mathbf{w}_{t+1}=\mathbf{w}_t-\eta_t\nabla\ell(\mathbf{w}_t;Z_{i_t}),$$

where i_t is sampled from $\{1,\dots,n\}$. To study stability, Hardt et al. [11] couple two runs on neighboring datasets S and $S^{(i)}$ using the same randomness and track the distance

$$\Delta_t=\mathbb{E}_\xi\|\mathbf{w}_t-\mathbf{w}_t^{(i)}\|. \tag{8}$$

The perturbation can enter the recursion only when the sampled index coincides with the replaced example, which happens with probability $1/n$. Under Lipschitz and smoothness assumptions, this gives the generic one-step estimate

$$\Delta_{t+1}\le\rho_t\Delta_t+\frac{2L\eta_t}{n}, \tag{9}$$

where ρ_t describes the expansiveness of the gradient update map. Thus, stability analysis reduces to understanding whether the optimization dynamics contract, preserve, or amplify the perturbation introduced by a single training example.

This recursion immediately reveals the different roles of convexity. For convex and β-smooth losses with $\eta_t\le 2/\beta$, the gradient update is non-expansive, so $\rho_t\le 1$. Hence,

$$\Delta_T\le\frac{2L}{n}\sum_{t=1}^{T}\eta_t,\qquad \epsilon_{\text{stab}}\le\frac{2L^2}{n}\sum_{t=1}^{T}\eta_t. \tag{10}$$

For λ-strongly convex losses, the update is contractive. The accumulated perturbation is then geometrically damped, leading to a training-time-independent rate of order $\epsilon_{\text{stab}}=\mathcal{O}(1/\lambda n)$. By contrast, in the nonconvex smooth case, the update map can be expansive, with a typical factor $\rho_t\le 1+\beta\eta_t$. The perturbation bound takes the form

$$\Delta_T\le\frac{2L}{n}\sum_{t=1}^{T}\eta_t\prod_{s=t+1}^{T}(1+\beta\eta_s),$$

which explains why nonconvex stability bounds usually have a weaker dependence on the training horizon. For polynomially decaying stepsizes $\eta_t\le c/t$, this mechanism yields a bound of order $\epsilon_{\text{stab}}=\mathcal{O}\left(T^{\frac{\beta c}{\beta c+1}}/n\right)$. These estimates summarize the basic message of Hardt et al. [11]: SGD generalizes when the perturbation injected by one training example is sufficiently averaged by random sampling and sufficiently controlled by the update dynamics. Convexity gives non-expansiveness, strong convexity gives contraction, while nonconvexity may create expansion. Representative bounds for SGD and related stochastic-gradient methods are summarized in Table III.

Building on this foundation, subsequent work refined the assumptions and limitations of the stability analysis of SGD

TABLE II
REPRESENTATIVE WORKS ON GENERALIZATION ANALYSIS OF RANDOMIZED OPTIMIZATION ALGORITHMS.

Theoretical tool	Important work	Other related work	Main role in this survey
Stability Analysis	[6] (JMLR, 2002), [11] (ICML, 2016), [12] (ICML, 2018), [13] (ICML, 2020)	[14] (NIPS, 2018), [15] (COLT, 2019), [16] (NIPS, 2020), [17] (COLT, 2020), [18] (ML, 2022), [19] (ICML, 2023) [20] (COLT, 2018), [21] (NIPS, 2021), [22] (ICML, 2022) [23] (AAAI, 2021), [24] (TPAMI, 2025), [25] (JMLR, 2026)	Sample-perturbation propagation along optimization trajectories
Information-theoretic	[26] (AIS, 2016), [27] (NIPS, 2017), [28] (ISIT, 2018) [29] (NIPS, 2019)	[30] (SAIT, 2020), [31] (NIPS, 2020), [32] (COLT,2021), [33] (NIPS 2021), [34] (COLT, 2023), [35] (NIPS, 2023) [36] (NIPS, 2021), [37] (COLT, 2023), [38] (NIPS, 2023) [39] (ITDS, 2021), [40] (NIPS, 2023), [41] (ICML, 2025)	Data-output dependence control via information measures
PAC-Bayesian	[42] (NIPS, 2017), [43] (NIPS 2018)	[44] (JMLR, 2012), [45] (NIPS, 2020), [46] (ICML, 2020) [47] (NIPS, 2018), [48] (NIPS, 2022), [49] (COLT, 2025)	Prior–posterior complexity control for randomized predictors
Algorithm-dependent complexity	[50] (NIPS, 2022), [51] (COLT, 2023)	[52] (ICML, 2018), [53] (NIPS, 2021), [54] (NIPS, 2021), [55] (NIPS, 2021), [56] (NIPS, 2022), [57] (NIPS, 2023)	Effective-class complexity control for training algorithms

TABLE III
REPRESENTATIVE STABILITY BOUNDS FOR SGD.

References	L-Lipschitz	β-Smooth	Convexity	Learning Rate	Results
Hardt et al. [11]	✓	✓	Convex	$\eta_t \leq 2/\beta$	$\mathcal{O}(L^2 \sum_{t=1}^{T} \eta_t/n)$
	✓	✓	λ-Strongly Convex	$\eta_t \leq 1/\beta$	$\mathcal{O}(L^2/(\lambda n))$
	✓	✓	Nonconvex	$\eta_t \leq c/t$	$\mathcal{O}(T^{\frac{\beta c}{\beta c+1}}/n)$
Kuzborskij et al. [12]	✓	✓	Convex	$\eta_t \leq 1/\sqrt{t}$	$\mathcal{O}(\sqrt[4]{T}/n + \sigma\sqrt{T}/n)$
	✓	✓	Nonconvex	$\eta_t \leq c/t$	$\mathcal{O}((L_\mu(w_1) \cdot T)^{\frac{\beta' c}{\beta' c+1}}/n)$
Bassily et al. [16]	✓	✗	Convex	$\eta_t = \eta$	$\mathcal{O}(\eta\sqrt{T} + \eta T/n)$
Lei et al. [13]	✗	✓	Convex	$\eta_t \leq 1/(2\beta)$	$\mathcal{O}(1/n)^\dagger$
	✗	✗	Convex	$\eta_t = cT^{-(3-\alpha^2-2\alpha)/4}$	$\mathcal{O}(n^{-\frac{1+\alpha}{2}})^\dagger$
Lei et al. [72]	✗	✓	Nonconvex	$\eta_t \asymp 1/(\gamma t)$	$\mathcal{O}(1/(n\gamma))^\dagger$
Zhang et al. [73]	✓	✓	Nonconvex	$\eta_t \leq c/(\beta t)$	$\mathcal{O}(T^c/n^{1+c})$

✓ means that the bound relies on the corresponding assumption, whereas ✗ means that the assumption is not required or has been removed. Here, σ denotes the gradient-noise variance, α is the Holder Smooth parameter, γ is the PL parameter, and β' is an initialization-dependent curvature parameter. The table only reports expected generalization bounds for SGD. Bounds marked with † are excess generalization bounds; all other bounds are standard generalization bounds.

in [11]. Bassily et al. [16] extended uniform stability to nonsmooth convex losses, showing that SGD can still achieve dimension-free generalization bounds under suitable step-size control even without smoothness, as in ReLU or hinge-loss models. Zhang et al. [73] studied the tightness of these results and showed that, within data-independent analyses, the bounds of Hardt et al. for convex and strongly convex objectives are essentially unimprovable without additional structure. For nonconvex losses, however, existing results remain more conservative, motivating a shift from unified worst-case bounds toward finer information about data, geometry, and training trajectories.

A key development along this direction is data-dependent and average model stability. Kuzborskij and Lampert [12] showed that the stability of SGD depends not only on step sizes and training time, but also on the risk and local geometry near initialization. In convex problems, smaller initial risk improves stability; in nonconvex problems, local curvature around initialization becomes decisive. This moves stability analysis beyond global Lipschitz and smoothness constants toward problem- and trajectory-dependent characterizations. Lei and Ying [13] further proposed average model stability, shifting the focus from loss differences to parameter differences. Their framework relates stability to a weighted sum of empirical risks along the SGD trajectory, avoids an explicit dependence on global gradient boundedness, and explains how low-noise conditions and better optimization can yield a fast $\mathcal{O}(1/n)$ generalization rate. Deng et al. [68] introduced locally elastic stability, which uses distribution-dependent local perturbation sensitivity to obtain tighter bounds than classical uniform stability. Together, these works mark a transition from worst-case stability to data-dependent, trajectory-aware, and distribution-aware formulations.

Later work incorporated additional geometric and statistical structure. Lei and Ying [72] studied nonconvex problems satisfying the Polyak–Łojasiewicz (PL) condition and decomposed the excess generalization error of SGD into an $\mathcal{O}(1/(n\gamma))$ generalization term and an optimization term, where γ is the PL constant. This suggests that, under favorable geometry, reducing training error need not worsen generalization, even in overparameterized regimes. Zhou et al. [18] introduced the average variance of stochastic gradients to derive high-probability bounds and explain the degradation caused by random label noise. Raj et al. [19] studied heavy-tailed gradient noise through stochastic differential equation approximations, revealing a potentially non-monotone relation between the tail index and generalization error. These results show that modern SGD stability analysis increasingly incorporates gradient variance, tail behavior, and local geometry, rather than relying only on step sizes, training time, and smoothness.

Overall, the stability analysis of SGD has evolved along a clear path: from the uniform-stability framework of Hardt et al. [11], to nonsmooth settings and tightness results, and then to data-dependent, average-model, local-geometric, and gradient-statistical refinements. Stability theory has therefore shifted from a coarse worst-case tool governed by global constants to a more refined framework aligned with stochastic optimization dynamics. This makes SGD the central object through which stability theory explains the generalization behavior of randomized optimization algorithms.

C. Algorithmic Variants and System-Level Stability

Beyond standard SGD, stability analysis has been extended to a broad range of randomized optimization algorithms and large-scale training mechanisms. Although these methods

differ in update rules, memory structures, noise injection, or communication protocols, their stability analyses share a common form. If Δ_t denotes the perturbation between two coupled runs on neighboring datasets, many results can be viewed as refinements of

$$\Delta_{t+1} \leq a_t \Delta_t + b_t/n + r_t, \tag{11}$$

where a_t measures the propagation of past perturbations, b_t/n is the new sample-level perturbation, and r_t captures algorithm-specific effects such as momentum memory, gradient noise, delay, consensus error, or zeroth-order estimation error. Thus, the main question is not whether each variant has a separate stability theory, but how the variant modifies the three terms in Eq. (11).

A first group of results isolates the effect of sampling and update randomness. Full-batch gradient descent removes mini-batch noise and therefore provides a useful baseline for understanding the role of stochasticity. Richards and Kuzborskij [74] studied average stability in overparameterized shallow networks through shortest gradient-descent paths, while Lei et al. [75] improved iterate-norm estimates and extended the analysis to SGD. For smooth convex losses, Nikolakakis et al. [76] further showed that full-batch gradient descent can enjoy tighter generalization bounds than SGD. These works clarify that stochastic sampling is not always benign: it may improve optimization efficiency, but it also introduces additional perturbation channels.

Mini-batch SGD modifies the sampling term b_t/n in Eq. (11). Since each update averages several sample gradients, the perturbation caused by replacing one example can be diluted by the batch. Lei et al. [77] studied average model stability of mini-batch SGD in convex, strongly convex, and nonconvex settings, showing that mini-batching can yield a linear speedup in generalization with respect to the batch size while preserving optimal excess generalization rates. This suggests that mini-batching affects generalization not only through computational parallelism, but also through the stability structure of the update.

Momentum and recursive gradient estimators mainly change the propagation factor a_t. Momentum introduces memory, so a perturbation created at one iteration may persist through future updates. Ramezani-Kebrya et al. [78] showed that stochastic momentum gradient descent can become unstable under multiple passes over the data and proposed early momentum to recover uniform stability under broader stepsize regimes. Pan et al. [79], [80] further showed that in STORM and related multi-level stochastic algorithms, deeper recursion can accumulate stochastic gradient variance and enlarge generalization error. These results indicate that momentum-like memory is not only an optimization device; it also changes how perturbations are stored and propagated.

Explicit-noise methods modify the recursion in a different way. Langevin algorithms inject external noise, which can smooth the dependence of the output on individual samples. SGLD [81] is the canonical example. Mou et al. [20] used squared Hellinger distance to prove uniform stability of SGLD in nonconvex settings, showing that Gaussian noise can reduce sensitivity to sample perturbations. Banerjee et al. [22] extended this view to exponential-family Langevin dynamics and obtained expected stability bounds controlled by gradient differences rather than global gradient norms. Thus, explicit noise is not merely an optimization perturbation; it can also serve as a stability-enhancing mechanism.

Other variants reshape the trajectory or the gradient estimator. LookAhead [82] uses periodic slow-weight updates, and Zhou et al. [83] showed that this smoothing of the parameter path can reduce trajectory divergence under data perturbations. SAM and related flatness-aware methods [84] incorporate local geometric information into the update; recent stability analyses suggest that LookAhead, SAM, and renormalization strategies may improve generalization by combining trajectory smoothing with flatness-inducing biases [85]–[89]. Variance-reduced methods, coordinate updates, and zeroth-order algorithms provide further examples: SVRG can improve stability by reducing gradient variance [90]; randomized coordinate descent can benefit from sparse updates in high dimensions [91]; and zeroth-order random search admits stability guarantees even without first-order gradients [25], [92], [93]. These examples show that stability analysis extends beyond vanilla stochastic gradients to a broad class of randomized update mechanisms.

System-level extensions introduce additional perturbation sources. In asynchronous, decentralized, and distributed optimization [23], [94]–[97], the algorithm is affected not only by sample replacement, but also by delay, communication topology, local model inconsistency, and data heterogeneity. A generic decomposition for decentralized methods is

$$\Delta_{t+1} \lesssim \Delta_t + \frac{\eta_t L}{mn} + \eta_t E_t, \tag{12}$$

where m is the number of agents and E_t represents a consensus or network-disagreement term. The first two terms resemble centralized SGD, whereas E_t carries the effect of topology and communication.

For asynchronous SGD [98], stability bounds must account for delayed gradients. Regatti et al. [95] obtained uniform-stability bounds for smooth nonconvex objectives with explicit delay dependence, while Deng et al. [99] used average model stability to remove global Lipschitz assumptions and obtain tighter nonvacuous bounds. Deng et al. [24] further derived exact stability recursions for quadratic problems through generating functions, showing that delay can be less harmful, and sometimes even regularizing, under suitable geometry and stepsizes.

For decentralized SGD [100], the central issue is how topology affects consensus and hence stability. Richards et al. [96] obtained topology-independent bounds for a representative D-SGD scheme, whereas Sun et al. [23] and Zhu et al. [101] showed that, for other update rules or weaker settings, the spectral gap can enter explicitly. Later work clarified that this dependence is not absolute. Bars et al. [102] recovered topology-independent rates under suitable topology-dependent stepsize conditions, and Wang et al. [103] extended the analysis to decentralized mini-batching, sampling without replacement, and zeroth-order optimization. Recent refinements further address relaxed assumptions, data heterogeneity,

time-varying topologies, and multi-round gossip communication [104]–[107].

When asynchrony and decentralization are combined, stability must control both temporal delay and spatial inconsistency. Deng et al. [97] derived stability-based generalization bounds for asynchronous decentralized SGD [108], showing that weaker connectivity and larger delay enlarge the stability bound in addition to the usual dependence on sample size, stepsize, and training time. Overall, system-level stability extends the classical SGD question from "how does a sample perturbation propagate through updates?" to "how do sample perturbations interact with delay, topology, consensus, and communication?"

D. Stability in Structured Learning Problems

The previous subsection focuses on changes in the optimization mechanism. Another line of work studies settings where the learning problem itself has a richer stochastic structure. In these problems, replacing one sample may not affect only one empirical loss term. A useful abstraction is

$$F_S(\mathbf{w}) = \frac{1}{|\mathcal{I}_S|} \sum_{\alpha \in \mathcal{I}_S} \phi(\mathbf{w}; Z_\alpha), \tag{13}$$

where α may index a single sample, a pair, a triplet, a task, or a client-level structure. The effect of replacing Z_i is then governed by

$$m_i = \big|\{\alpha \in \mathcal{I}_S : i \in \alpha\}\big|, \tag{14}$$

namely the number of objective terms in which Z_i participates. Thus, structured stability must track not only the optimizer's sensitivity, but also the structural multiplicity through which one sample enters the objective.

Dependent sampling is the first departure from the standard i.i.d. setting. In Markov-chain-sampled SGD, the stochastic gradients are biased and correlated across time. Wang et al. [109] showed that, with suitable stepsize decay, MC-SGD can still achieve optimal excess risk comparable to i.i.d. SGD. The key message is that temporal dependence need not destroy stability, but it changes the perturbation recursion through mixing and bias terms.

Stochastic compositional optimization introduces nested expectations and multi-level randomness. Yang et al. [110] developed compositional uniform stability for SCO algorithms, and Chen et al. [111] extended the analysis to zeroth-order SCO. In this case, perturbations propagate not only through outer stochastic gradients, but also through inner stochastic estimates. The stability notion must therefore reflect the coupling between the inner and outer layers of the objective.

Minimax and adversarial learning introduce coupled optimization dynamics. For problems of the form

$$\min_x \max_y \; F_S(x, y), \tag{15}$$

a sample perturbation can be amplified through both the minimization and maximization variables. Farnia and Ozdaglar [112] used stability to compare GDA, GDmax, and PPM, showing that solving the inner maximization more accurately does not necessarily improve generalization. Lei et al. [113] and Zhang et al. [114] developed stability bounds for stochastic minimax and saddle-point problems, including optimal sample-size-dependent rates under convex–concave or strongly convex–strongly concave structures. Ozdaglar et al. [115] further argued that the primal gap can be a more appropriate generalization measure in nonconvex minimax problems. For adversarial training, stability analyses of nonsmooth adversarial losses, smoothing methods, and PGD-type perturbation maps show that robust generalization depends critically on how adversarial dynamics amplify sample-level perturbations [116]–[119].

Pairwise and higher-order learning provide a clear example of structural multiplicity. For a pairwise objective,

$$F_S(\mathbf{w}) = \frac{1}{n(n-1)} \sum_{i \neq j} \ell(\mathbf{w}; Z_i, Z_j), \tag{16}$$

replacing one example affects $O(n)$ loss terms rather than one. Lei et al. [120], [121] established stability and generalization bounds for SGD in pairwise learning, including nonsmooth and nonconvex objectives. Wu et al. [122] studied randomized coordinate descent for pairwise learning and obtained an $\mathcal{O}(1/n)$ expected generalization bound under low-noise conditions. Wang et al. [123] and Chen et al. [124] extended stability analysis to point-pair and triplet learning. These works show that stability can handle non-pointwise objectives once the sample-replacement structure is explicitly accounted for.

Meta-learning adds task-level randomness and bilevel optimization. A typical meta-learning problem contains both within-task samples and a distribution over tasks, so generalization depends on perturbations at two levels. Maurer [125] first used uniform stability to analyze meta-algorithms. Al-Shedivat et al. [126] studied MAML and Reptile and showed that different inner-loop procedures lead to different stability behavior. Farid et al. [127] combined base-level stability with meta-level PAC-Bayes analysis. Later work refined the rates under strong convexity or milder assumptions [128], [129], and more recent studies developed uniform meta-stability and compared gradient-based with proximal-descent frameworks [130], [131]. These results show that stability in meta-learning must account for both within-task perturbations and the transfer mechanism induced by the outer-level update.

Federated learning introduces client-level structure. Unlike decentralized SGD, where topology is often the primary issue, federated learning is dominated by data heterogeneity, partial participation, local training, and server aggregation. Stability analyses of FedAvg, FedProx, SCAFFOLD, and related methods show that generalization depends on the interaction between local update steps, client distribution mismatch, and aggregation. Chen et al. [132] established guarantees for federated zeroth-order optimization, while Liu et al. [133] showed that in personalized federated learning, the decoupling of shared and personalized variables and the communication topology can substantially affect generalization. Thus, federated stability must jointly control sample perturbations, client heterogeneity, and communication structure.

Overall, structured stochastic learning extends stability from simple sample-level perturbations to more complex perturbation pathways. In dependent sampling, perturbations propagate

through temporal correlation; in compositional optimization, through nested stochastic estimates; in minimax learning, through coupled primal-dual dynamics; in pairwise and triplet learning, through higher-order sample relations; in meta-learning, through task-level transfer; and in federated learning, through client heterogeneity and aggregation. These settings differ technically, but they share the same principle: stability remains useful once the structure through which a single sample influences the learning objective is made explicit.

E. Discussion and Limitations

The results reviewed above show that stability has developed from a generalization tool for ERM into one of the most systematic frameworks for randomized optimization. Its strength is that it directly follows the training dynamics: a one-sample perturbation is introduced into the update rule, and the analysis tracks how this perturbation is damped, amplified, or reorganized by the algorithm.

This makes stability particularly suitable for SGD and its variants. Step sizes, training time, batch size, momentum, explicit noise, delay, communication topology, and client heterogeneity can all enter the generalization bound through the stability recursion. The literature has therefore moved from uniform stability to average model stability, data-dependent stability, local and distribution-dependent stability, and high-probability stability. These refinements reduce the conservativeness of worst-case Lipschitz and smoothness constants by incorporating trajectory information, local geometry, gradient statistics, and system structure.

At the same time, stability is not a complete theory of generalization. Technically, each new algorithmic or structural setting often requires a new perturbation recursion, which can become highly problem specific. Conceptually, stability measures sensitivity to data perturbations; it does not directly quantify posterior complexity, information leakage, or the effective size of the class explored by training. In nonconvex and overparameterized regimes, stability bounds may still be conservative unless supplemented by local geometry, noise structure, or additional algorithm-dependent information.

Thus, stability is the most mature route for connecting randomized optimization trajectories to generalization, but it should be viewed as one piece of a broader picture. Information-theoretic methods describe data-output dependence, PAC-Bayes analyzes prior-posterior complexity, and algorithm-dependent complexity studies the effective class induced by training. The following sections develop these complementary perspectives.

IV. Information-theoretic analysis

The information-theoretic perspective views a learning algorithm as a stochastic channel from the training sample to the learned model, rather than merely as a deterministic map from data to parameters. In this view, generalization is controlled not primarily by the size of the hypothesis space or by the output difference on neighboring datasets, but by the statistical dependence between S and W. The central question is: how much information about the training sample is retained in the final model, and how does this retained information affect performance on unseen data? Entropy, Kullback–Leibler divergence, and mutual information provide a natural language for this question, as they quantify uncertainty, distributional shift, and dependence in a unified way [134]–[140].

The basic dependence measure is the input-output mutual information

$$I(S;W) = D_{\mathrm{KL}}(P_{S,W} \,\|\, P_S \otimes P_W)\,. \tag{17}$$

If $I(S;W)$ is small, then the output model depends only weakly on the particular training sample, suggesting that its empirical performance is less likely to be sample-specific. This is the main contrast with stability analysis: stability asks whether replacing one sample significantly changes the output, whereas information theory asks how much information about the entire sample is encoded in the output.

Modern information-theoretic generalization analysis largely began with Russo and Zou [26], who showed in adaptive data analysis that the bias of an adaptively chosen statistic can be controlled by the amount of information used. Xu and Raginsky [27] then extended this idea to general learning algorithms. For example, if the loss is σ-sub-Gaussian under the appropriate product distribution, their result gives the canonical bound

$$\left|\mathbb{E}_{S,W}\left[L_\mu(W) - L_S(W)\right]\right| \le \sqrt{\frac{2\sigma^2}{n} I(S;W)}.$$

This result made precise the principle that information controls generalization and established a line of work parallel to uniform convergence and algorithmic stability. However, the global quantity $I(S;W)$ is often too coarse for modern randomized training. It compresses all dependence between the training sample and the output into a single scalar, and therefore cannot reveal which samples, which updates, or which parts of the trajectory carry the most information. Moreover, in iterative stochastic algorithms, generalization is often governed not by the total input-output dependence alone, but by local, conditional, or samplewise dependence accumulated during training. This motivated finer information measures. Bu et al. [30] introduced individual-sample mutual information, replacing $I(S;W)$ by quantities such as $I(Z_i;W)$ and obtaining bounds of the form

$$\left|\mathbb{E}_{S,W}\left[L_\mu(W) - L_S(W)\right]\right| \le \frac{1}{n}\sum_{i=1}^{n} \sqrt{2\sigma^2 I(Z_i;W)}. \tag{18}$$

Haghifam et al. [141] further developed conditional mutual information and related decompositions, moving the theory toward supersample and leave-one-out frameworks. Thus, the information-theoretic route evolved from a global dependence bound into a localized framework capable of reflecting more detailed training structures.

For this survey, the most relevant question is how these ideas apply to randomized optimization algorithms. Such algorithms are multi-step, noisy, and path-dependent random mappings. Hence, a single input-output term $I(S;W)$ is often insufficient. The key development is to reinterpret training as a dynamic information-transmission process: information about the data

is injected, propagated, attenuated, or amplified along the optimization trajectory.

A. From Mutual Information to Randomized Optimization

The connection between information theory and randomized optimization becomes explicit when training is viewed as a composition of stochastic channels:

$$S \longrightarrow W_0 \longrightarrow W_1 \longrightarrow \cdots \longrightarrow W_T.$$

Instead of bounding only $I(S;W_T)$, one can use the chain rule to study the information accumulated along the path:

$$I(S;W_{0:T}) = \sum_{t=0}^{T-1} I(S;W_{t+1} \mid W_{0:t}). \tag{19}$$

This formulation is especially natural for noisy iterative algorithms, where each update can be regarded as a noisy channel.

Pensia et al. [28] made this connection precise for a broad class of noisy iterative algorithms with bounded updates and Markovian structure. Their framework applies to SGLD, SGHMC, and related variants, and covers the last iterate, tail averaging, and more general path-dependent outputs. Compared with the global input-output bound of [27], this work brought information-theoretic analysis directly into stochastic optimization dynamics. SGLD then became a representative example. Its update can be written as

$$W_{t+1} = W_t - \eta_t \nabla L_S(W_t) + \sqrt{2\eta_t/\beta}\,\zeta_t,\ \zeta_t \sim \mathcal{N}(0,I),$$

where the injected Gaussian noise plays a dual role: it drives the stochastic dynamics and limits the information that each update can reveal about the training sample. In Gaussian-channel analyses, the stepwise information is controlled by the signal-to-noise ratio of the update. Schematically, one obtains terms of the form

$$I(S;W_{0:T}) \lesssim \sum_{t=0}^{T-1} \frac{\eta_t^2}{\sigma_t^2}\mathbb{E}\big[\|g_t(S) - g_t(S')\|^2\big], \tag{20}$$

where the numerator reflects data-dependent gradient variation and the denominator reflects the strength of injected noise.

Several works sharpened this picture. Negrea et al. [29] introduced data-dependent estimates for SGLD, showing that suitable reference measures can yield substantially tighter and more problem-dependent bounds. Wang et al. [36] analyzed SGLD through Gaussian channels, further emphasizing that the noise structure is not a technical artifact but a key mechanism governing information flow. Farghly and Rebeschini [33] and Futami and Fujisawa [35] derived time-independent information-theoretic bounds for nonconvex SGLD, shifting attention from whether such bounds exist to whether they necessarily deteriorate with training time. Thus, SGLD is central in this section not because it is the most widely used optimizer, but because it most clearly embodies the information-theoretic intuition: algorithmic noise simultaneously shapes optimization and suppresses the accumulation of data information.

The information-theoretic route was later extended beyond explicitly noise-injected algorithms. Neu et al. [32] showed that, for SGD under general nonconvex losses, generalization can be controlled by local quantities along the trajectory, including stochastic-gradient variance, local smoothness, and the sensitivity of the final loss to perturbations. This shows that information theory is not restricted to Langevin-type methods. With suitable modifications, it can also describe standard stochastic-gradient training. Nevertheless, compared with stability analysis, information-theoretic analysis of SGD remains less developed in breadth [32], [41].

Overall, this line underwent a clear transition. Early bounds measured global dependence between S and W. Noisy iterative analyses decomposed this dependence across training steps. For SGLD and SGD, the stepwise terms were further related to noise structure, gradient statistics, and local geometry. The main contribution of information theory to randomized optimization is therefore not merely a new family of bounds, but a reinterpretation of training as a dynamic information-transmission process.

B. Finer Information Measures

Although $I(S;W)$ provides a clean and general bound, it is often too coarse to explain modern training. It does not indicate which training examples are most influential, nor does it distinguish global memorization from localized dependence. This motivated finer information quantities. The first refinement is individual-sample mutual information. Bu et al. [30] replaced the global term $I(S;W)$ by samplewise quantities $I(Z_i;W)$, leading to bounds such as (18). This refinement is important not only because it can be tighter, but also because it moves the analysis from global input-output dependence to samplewise dependence. In this sense, information-theoretic generalization becomes closer in spirit to stability, although it measures dependence rather than perturbation sensitivity.

Conditional mutual information further localizes the analysis. In the supersample framework, one draws paired samples

$$\widetilde{S} = \{(Z_i^0, Z_i^1)\}_{i=1}^n, \qquad U_i \sim \mathrm{Bernoulli}(1/2),$$

and forms the training sample by selecting $S = \{Z_i^{U_i}\}_{i=1}^n$. The relevant quantity is then not only $I(S;W)$, but the conditional dependence between the output and the selection variables: $I(W;U \mid \widetilde{S})$. Haghifam et al. [31] showed that such conditional and disintegrated information measures can yield sharper bounds and can be applied to noisy iterative algorithms. The intuition is that the supersample provides a reference environment, while the selection variables encode the sample-replacement structure. Hence, the information quantity becomes closer to leave-one-out analysis.

Leave-one-out conditional mutual information makes this connection even more explicit. Issa et al. [34] showed that, for bounded losses, leave-one-out-type information quantities can control average generalization and connect information-theoretic bounds with classical leave-one-out risk analysis in certain interpolating algorithms. A representative form is

$$I_{\mathrm{LOO}} = \frac{1}{n}\sum_{i=1}^{n} I(W;U_i \mid \widetilde{S}, U_{-i}), \tag{21}$$

which measures how much the output reveals about the inclusion of each individual example after conditioning on the remaining selection variables.

These developments show that the information-theoretic route is not a single theory centered only on $I(S;W)$. Rather, it is an evolving family of tools that progressively refine the dependence measure: from global mutual information, to individual-sample information, to conditional and leave-one-out information. This refinement is especially important for randomized optimization. In SGLD, dependence is reshaped by injected noise at each step; in SGD, it is encoded in local gradient statistics along the trajectory. In both cases, generalization is often governed not by total input-output dependence alone, but by how information is accumulated and redistributed during training.

C. Discussion and Limitations

Overall, the information-theoretic route provides a perspective on the generalization of randomized optimization algorithms that is markedly different from both stability and capacity-based approaches. It neither tracks the size of the hypothesis space directly nor compares the outputs produced on neighboring training sets. Instead, it interprets the training process as a stochastic information-transmission procedure from samples to model, and studies to what extent the training data are "encoded" into the final output. This perspective is particularly effective for noisy iterative algorithms, SGLD, and part of the SGD literature, because such algorithms themselves already involve explicit noise injection or stochastic sampling mechanisms, and can therefore be naturally viewed as iterative compositions of noisy channels. For this reason, although the information-theoretic line does not cover as broad a range of optimizers as stability, it does form a relatively clear internal trajectory in the sequence noisy iterative algorithms $\rightarrow$ SGLD $\rightarrow$ SGD, with an evident progression in both methodology and scope.

At the same time, this route also has clear limitations. The central issue is that the fact that an information quantity admits a theoretically meaningful upper bound does not imply that it necessarily captures the statistical mechanism that truly determines generalization. Haghifam et al. [37] systematically studied the limitations of several information-theoretic generalization techniques in stochastic convex optimization, and showed that even when such bounds are formally tight, they are still insufficient to recover minimax-optimal rates for gradient descent methods. Moreover, even if one artificially constructs a noisy surrogate by adding Gaussian noise to the algorithm output, this limitation does not disappear. This suggests that the difficulty is not merely that the constants are loose, but rather that the chosen dependence measure itself may be structurally misaligned with the actual generalization mechanism of the problem. The work of Livni [38] further reinforces this point from a lower-bound perspective: some families of information-theoretic upper bounds possess unavoidable lower-bound limitations of their own. The problem is therefore not only how to sharpen a bound, but also which notion of information truly corresponds to the factors that govern generalization during training.

It is precisely in this context that recent work has begun to combine information-theoretic tools with other theoretical frameworks [39], [40] in order to mitigate the limitations of pure information control. For example, sample-conditioned hypothesis stability has been used to sharpen information-theoretic generalization bounds, indicating that stability and information theory are not simply two parallel routes, but can intersect at a finer level of dependence control. For the purposes of this survey, this point is particularly important, because it shows that while the information-theoretic perspective can describe training through the language of information flow, it is often insufficient on its own once the true training structure becomes too complex and global mutual information, or even its localized variants, remains too coarse. In such cases, stability or other frameworks are often still needed in order to obtain conclusions that are both tighter and more interpretable.

From the perspective of the overall structure of this survey, the strengths and limitations of the information-theoretic route are therefore both fairly clear. It is particularly well suited to answering questions such as: to what extent does a training algorithm retain information about the training data, how does noise suppress the accumulation of such information, and how do local statistical quantities along the training trajectory influence the eventual generalization behavior? What it is less well suited for are situations in which the mechanism that truly determines generalization is not primarily reflected in input-output information coupling. For this reason, the information-theoretic perspective is best understood here as a route that complements stability, but does not always provide a complete explanation on its own. It offers a very natural information-flow view of randomized optimization algorithms, while its limitations also serve as a reminder that the generalization mechanisms of modern training procedures are often too rich to be fully captured by any single theoretical language.

V. PAC-Bayesian Analysis

PAC-Bayes theory provides a posterior-based route to generalization. Instead of measuring the worst-case complexity of the whole hypothesis space, it evaluates the complexity of a learned distribution Q relative to a reference prior P. A representative PAC-Bayes bound has the form

$$L_\mu(Q) \lesssim L_S(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(1/\delta)}{n}}, \qquad (22)$$

which holds with high probability over the training sample, up to constants and the precise choice of PAC-Bayesian inequality. Thus, the key quantity is not the size of the parameter space itself, but the prior-posterior deviation $\mathrm{KL}(Q\|P)$. This is the sense in which PAC-Bayes keeps the finite-sample, high-probability spirit of PAC learning while introducing a Bayesian prior-posterior language.

The earliest representative PAC-Bayesian bounds were developed by McAllester et al. [142], [143], who showed that a posterior distribution with small empirical risk and small prior-posterior divergence enjoys controlled population risk. Seeger et al. [144] connected this framework to Bayesian learning settings such as Gaussian process classification, while

Catoni [145] further developed localized and Gibbs-posterior viewpoints. Subsequent work enriched both the theory and its practical scope, including data-dependent priors [44], sample-dependent priors [45], nonvacuous PAC-Bayes bounds for deep networks [146], and PAC-Bayesian analyses of majority votes or Gibbs-type predictors [147]–[149].

In this survey, however, we do not aim to review PAC-Bayes theory in full generality. Our focus is narrower: how does PAC-Bayes interact with the generalization analysis of randomized optimization algorithms? Compared with stability, PAC-Bayes does not directly track the propagation of sample perturbations along neighboring trajectories. Compared with information-theoretic analysis, it does not primarily measure the dependence between S and W. Its main role is to provide a complexity language for the randomized objects induced by training:

$$\text{training} \Rightarrow Q \quad \text{and} \quad \mathrm{KL}(Q\|P) \Rightarrow \text{generalization}.$$

This makes PAC-Bayes particularly useful when a randomized optimizer induces a posterior distribution, a local perturbation neighborhood around the learned solution, or a data-dependent prior. In such settings, PAC-Bayes is often combined with stability, differential privacy, or sensitivity control to make these algorithm-induced priors or posteriors theoretically admissible.

This perspective also explains why PAC-Bayes is closely related to flatness, sharpness, and local geometry in modern deep learning: a flat region around a learned solution can often support a broader posterior without incurring a large complexity cost. These mechanism-level connections will be discussed later together with other modern generalization phenomena. In the present section, we focus on PAC-Bayes as a posterior-complexity interface between randomized training procedures and generalization guarantees.

A. PAC-Bayes and Randomized Learning

PAC-Bayes becomes directly relevant to randomized learning when the posterior is no longer viewed as an abstract distribution chosen after training, but as an object induced by the learning algorithm itself. For a randomized optimizer, training may generate a distribution through random initialization, mini-batch sampling, random hyperparameters, posterior perturbations, or adaptive choices made during optimization. Thus, the algorithmic output is better represented as

$$S \xrightarrow{A,\xi} Q, \qquad W \sim Q, \tag{23}$$

where Q denotes the posterior distribution associated with the training procedure. The key question is then not only how to bound the risk of a given posterior, but also how to justify the posterior produced by a concrete randomized algorithm.

London [42] provided an important step in this direction by studying PAC-Bayesian generalization bounds for randomized learning algorithms, with SGD as a representative example. The significance of this work is not that it gives a broad optimizer-by-optimizer theory comparable to stability analysis. Rather, it shows that algorithmic randomness itself can define a posterior-like object whose complexity can be controlled through PAC-Bayesian tools. Random sampling, random hyperparameters, and adaptive procedures can therefore be brought into the prior-posterior framework, rather than treated merely as external sources of randomness.

This connection also reveals a basic difficulty. Classical PAC-Bayes bounds require the prior to be independent of the training sample, while randomized optimization often suggests priors or posteriors shaped by the data, the trajectory, or the learned solution. In this case, PAC-Bayes alone is usually not enough. One needs an additional mechanism to control how strongly the algorithm-induced distribution depends on the data. This is where stability becomes a natural bridge:

$$\text{sensitivity control} + \mathrm{KL}(Q\|P) \implies \text{generalization control}.$$

Stability controls the effect of data perturbations, while PAC-Bayes measures the complexity of the induced posterior relative to a prior. The two tools therefore play complementary roles.

Rivasplata et al. [43] developed this connection by studying PAC-Bayesian bounds for stable algorithms with instance-dependent priors. Their results show that, under suitable stability control, valid risk estimates can still be obtained even when the prior is allowed to depend on the data-generating distribution. This point is especially important for randomized optimization: PAC-Bayes provides the posterior complexity term, but stability or related sensitivity controls often make algorithm-induced priors theoretically admissible.

Thus, the main interface between PAC-Bayes and randomized optimization can be summarized as follows. A randomized optimizer may not only return a point estimate w_T, but also induce a distribution around the learned solution. PAC-Bayes then asks whether this distribution has small empirical risk and small prior-posterior complexity. This perspective differs from stability: instead of tracking how two neighboring trajectories separate, PAC-Bayes characterizes the effective complexity of the posterior neighborhood produced by training.

B. Sample-Dependent Priors and Data-Dependent Priors

The choice of prior is one of the central issues in applying PAC-Bayes to modern training. A fully sample-independent prior is theoretically clean, but it may be too crude to reflect the local structure discovered during optimization. In contrast, a data-dependent prior can be much sharper, but it violates the classical requirement that the prior be independent of the sample. The tension can be expressed as

$$P \perp S \quad \text{vs.} \quad P_S = \mathcal{M}(S).$$

The central question is therefore how to use data- or sample-dependent priors without invalidating the PAC-Bayesian guarantee.

Awasthi et al. [45] studied PAC-Bayes bounds for randomized algorithms with sample-dependent priors. Their work treats the dependence of the prior on the sample as a primary object of analysis, rather than as a minor technical complication. Under suitable sensitivity conditions, sample-dependent priors can still lead to valid generalization bounds. This is

highly relevant for randomized optimization, since the initialization, optimization path, local curvature, and algorithmic statistics may all suggest informative reference distributions.

Dziugaite and Roy [150] provided another influential route through differential privacy. The basic idea is that a data-dependent prior can be used if the mechanism that constructs it reveals sufficiently little information about the training sample. Schematically,

$$S \xrightarrow{\mathcal{M}_{\mathrm{priv}}} P_S, \qquad \mathcal{M}_{\mathrm{priv}} \text{ is differentially private.}$$

Then the dependence of P_S on S can be controlled, and PAC-Bayesian bounds remain valid after appropriate correction. They also showed that even when the prior is not generated directly by a private mechanism, one can still obtain guarantees if its distance to a private mechanism is controlled. Their analysis of Gaussian priors with means induced by SGLD illustrates a key point: the training algorithm itself can participate in constructing the prior used in PAC-Bayes analysis.

This viewpoint also clarifies the role of Entropy-SGD. Entropy-SGD was originally proposed to bias optimization toward wide valleys through a local-entropy objective [151]. Dziugaite and Roy [47] showed that it can also be interpreted as optimizing a prior appearing in a PAC-Bayes bound. The subtlety is that this prior is constructed from the training data. Differential privacy therefore enters not mainly as a privacy goal, but as a device for legitimizing a data-dependent prior.

These developments change the role of PAC-Bayes in modern learning theory. In early PAC-Bayes, one typically analyzed a given prior-posterior pair. In randomized optimization, the more algorithmic question is whether the local structure generated by training can itself be organized into a valid PAC-Bayesian object:

$$\text{trajectory} \Rightarrow (P, Q) \Rightarrow \text{generalization}. \tag{24}$$

In this sense, PAC-Bayes becomes a bridge between training-induced randomness, data-dependent structure, and generalization.

C. From Posterior Design to Gradient Methods

Building on randomized learning and data-dependent priors, recent PAC-Bayesian analysis has moved in two related directions. The first aims to make PAC-Bayes bounds computable and nonvacuous for modern deep networks. The second brings PAC-Bayesian reasoning closer to gradient-based training procedures, rather than applying it only to abstract randomized predictors.

A useful way to summarize the modern PAC-Bayes objective is

$$\mathcal{B}_P(Q; S) = L_S(Q) + \psi_n(\mathrm{KL}(Q\|P) + \log(1/\delta)), \tag{25}$$

where $\psi_n(\cdot)$ denotes the complexity penalty appearing in the chosen PAC-Bayes inequality. For example, in square-root-type bounds, $\psi_n(x)$ scales as $\sqrt{x/n}$. Thus, PAC-Bayes analysis is not only about evaluating a posterior Q, but also about designing Q so that it achieves a favorable tradeoff between empirical risk and prior-posterior complexity.

This viewpoint is central to the work of Dziugaite and Roy [146], who showed that directly optimizing a PAC-Bayes bound can yield nonvacuous numerical guarantees for stochastic neural networks with far more parameters than training samples. In this case, the relevant object is not the whole parameter space, but a local posterior around a learned solution. More abstractly, if $\mathbf{w}$ is a trained parameter and ρ is a perturbation distribution, one may consider

$$Q_{\mathbf{w},\rho} = \mathrm{Law}(\mathbf{w} + \zeta), \qquad \zeta \sim \rho. \tag{26}$$

The empirical term then becomes

$$L_S(Q_{\mathbf{w},\rho}) = \mathbb{E}_{\zeta\sim\rho}\big[L_S(\mathbf{w} + \zeta)\big], \tag{27}$$

which measures how well the neighborhood around w performs on the training data. The complexity term $\mathrm{KL}(Q_{w,\rho}\|P)$ measures how costly this neighborhood is relative to the prior. This formulation makes posterior design a substantive part of the analysis: the center, scale, and shape of the perturbation distribution all affect the final bound.

This also explains the connection between PAC-Bayes and Entropy-SGD. Entropy-SGD was originally proposed to bias optimization toward wide valleys through a local-entropy objective [151]. Dziugaite and Roy [47] showed that such a procedure can also be interpreted through the optimization of a prior appearing in a PAC-Bayes bound. The key issue is that this prior may be constructed from the training data, so its use must be justified through additional tools such as differential privacy or sensitivity control. Thus, the modern PAC-Bayesian question is no longer only how to bound a fixed prior-posterior pair, but also how the training process constructs a meaningful and admissible prior-posterior structure.

A second direction brings PAC-Bayes closer to gradient methods themselves. For a randomized gradient method, the final iterate can be viewed as a random variable

$$W_T = \Phi_T(S, \xi), \qquad Q_T = P_{W_T|S}, \tag{28}$$

where ξ collects the randomness from initialization, sampling, or noise injection. From a PAC-Bayesian viewpoint, the law Q_T induced by the algorithm becomes the posterior-like object to be controlled. Luo et al. [48] developed this type of connection using discrete and continuous priors, deriving high-probability generalization bounds for Floored GD, Floored SGD, and GLD/SGLD, including nonconvex and nonsmooth settings. Compared with analyses that introduce a posterior only after training, this line ties the PAC-Bayesian object more directly to the optimization dynamics.

Recent work further shows that PAC-Bayesian reasoning can also be extended toward deterministic gradient dynamics. Clerico et al. [49] developed deterministic PAC-Bayes bounds for gradient descent and gradient flows and obtained computable guarantees without relying on an explicit derandomization step. In such settings, the posterior need not be interpreted only as external algorithmic randomness. It may also describe a local distribution or neighborhood associated with the deterministic training path.

PAC-Bayes is therefore naturally connected to local geometry. Tsuzuku et al. [46] used PAC-Bayesian reasoning to

study normalized flat minima and scale-invariant flatness. In the notation of (26)–(27), a flat solution is one for which $L_S(Q_{w,\rho})$ remains small even when ρ has relatively large spread. PAC-Bayes translates this perturbation tolerance into a complexity question: how broad can the posterior be while keeping both the empirical risk and $\mathrm{KL}(Q\|P)$ under control? A fuller discussion of flatness, sharpness, and local geometry is deferred to the final discussion, where these mechanisms can be compared with stability and algorithm-dependent complexity.

Overall, these developments show that PAC-Bayes has moved beyond its early role as a theory of randomized classifiers and Gibbs posteriors. It is increasingly connected to posterior design, data-dependent priors, nonvacuous deep-learning bounds, and gradient-based methods. For this survey, the main point is not that PAC-Bayes provides an optimizer-specific theory as broad as stability. Rather, it provides a posterior-complexity language for describing the local distributions and data-dependent structures induced by modern training algorithms.

D. Discussion and Limitations

The role of PAC-Bayes in randomized optimization is more specific than that of stability. Stability directly studies how a sample perturbation propagates along the optimization trajectory. PAC-Bayes instead studies whether the distributional object associated with training has small empirical risk and small prior-posterior complexity. In the notation of (25), its central question is whether one can find a posterior Q such that

$$L_S(Q) \text{ is small} \quad \text{and} \quad \mathrm{KL}(Q\|P) \text{ is controlled.}$$

This is why PAC-Bayes is particularly useful when the output of training is randomized, when the learned solution admits a meaningful local perturbation distribution, or when the training process suggests an informative data-dependent prior.

This perspective gives PAC-Bayes a clear but narrower position in the generalization analysis of randomized optimization. It does not usually provide detailed perturbation recursions for a wide range of optimizers. Instead, it answers a different question: can the random neighborhood, posterior distribution, or data-dependent prior induced by training be organized into a valid PAC-Bayesian object? When the answer is positive, PAC-Bayes provides a finite-sample, high-probability control of the corresponding randomized predictor through the empirical-risk–complexity tradeoff in Eq. (25).

The limitations are equally important. First, the quality of a PAC-Bayes bound depends heavily on the choice of P and Q. A bound may be valid but uninformative if the posterior does not align with the local geometry of the trained solution or if the prior is too crude. Second, the priors most useful in modern training are often data-dependent or algorithm-induced. Such priors require additional justification, typically through stability, differential privacy, or sensitivity control. Third, although PAC-Bayes has been connected to deep nonvacuous bounds and to several gradient-based methods, it has not yet developed into a broad optimizer-by-optimizer theory comparable to stability analysis.

For these reasons, PAC-Bayes is best viewed as a complementary posterior-complexity framework. Its direct algorithmic coverage is narrower than stability, but it captures an aspect of randomized optimization that stability does not directly describe: the effective complexity of the distributional structure around the learned solution. This makes PAC-Bayes indispensable for understanding randomized learning procedures whose outputs are naturally posterior-like, locally perturbed, or shaped by data-dependent priors.

VI. Algorithm-dependent complexity

Classical statistical learning theory often explains generalization through uniform convergence. Given a hypothesis class $\mathcal{H}$, one seeks a bound of the form

$$\sup_{h\in\mathcal{H}} |L_\mu(h) - L_S(h)| \le \varepsilon_n(\mathcal{H}), \tag{29}$$

where $\varepsilon_n(\mathcal{H})$ is controlled by VC dimension, covering numbers, or Rademacher complexity. If Eq. (29) is small, then empirical risk minimization over $\mathcal{H}$ generalizes. This principle has long served as the dominant capacity-control explanation in learning theory.

Modern overparameterized training challenges this global view. The issue is not only that classical bounds are numerically loose, but also that the object $\mathcal{H}$ may be too large to describe what the algorithm actually explores. Nagarajan and Kolter [152] showed that many uniform-convergence-type bounds remain vacuous even when restricted to classifiers produced by gradient descent in certain overparameterized models, and may empirically worsen as the sample size increases. This suggests that the difficulty is not merely a matter of constants; it concerns the object on which uniform control is imposed.

Subsequent works [153]–[155] therefore shifted the focus from the global class $\mathcal{H}$ to an algorithm-dependent effective object. A schematic form of this shift is $\mathcal{H} \rightsquigarrow \mathcal{H}_{A,S}$, where $\mathcal{H}_{A,S}$ denotes the class, trajectory, random set, or local structure actually induced by the training algorithm and the data. The goal is no longer to prove uniform convergence over the entire ambient class, but to establish a bound such as

$$\sup_{h\in\mathcal{H}_{A,S}} |L_\mu(h) - L_S(h)| \le \varepsilon_n(\mathcal{H}_{A,S}). \tag{30}$$

This section reviews several ways in which this effective object has been constructed: compression, surrogate predictors, localized covers, algorithm-dependent Rademacher complexity, random sets, and fractal or geometric complexity.

A. Compression, Surrogates, and Effective Description Length

One early response to the limitations of global uniform convergence is compression. Although a deep network may contain a very large number of parameters, the trained predictor may admit a much shorter description. Arora et al. [52] showed that nonvacuous generalization bounds can be obtained by explicitly compressing a trained network. In this view, the relevant complexity is not the nominal parameter count, but the description length of the compressed predictor. Schematically,

if $\mathrm{Comp}(h_S)$ denotes the number of bits needed to encode the compressed model, the resulting bound takes the form

$$L_\mu(h_S) \lesssim L_S(h_S) + \sqrt{\frac{\mathrm{Comp}(h_S) + \log(1/\delta)}{n}}. \tag{31}$$

Thus, compression replaces the global class complexity by the effective description length of the model actually reached by training.

A related response is based on surrogate predictors. Negrea et al. [154] did not simply reject the negative conclusions of Nagarajan and Kolter [152]; instead, they asked which object should satisfy uniform convergence. Their answer is that the original interpolating predictor may be the wrong object. By constructing a surrogate predictor coupled to the learner, one can impose uniform control on a more appropriate effective class. The generalization guarantee is then proved not by controlling the entire original class, but by relating the learner to a surrogate object for which capacity control becomes meaningful.

Compression and surrogate methods differ technically, but they share the same principle. Both preserve a capacity-control viewpoint while changing the object of analysis. Compression uses a short description of the trained model; surrogate methods replace the original predictor by a structurally modified object. In both cases, the key question is no longer how large the ambient hypothesis space is, but how complex the algorithm-relevant object is.

B. Localized Covers and Algorithm-Dependent Complexity

A more direct way to formalize this idea is to build complexity measures around the classes actually visited by the algorithm. Park et al. [50] proposed localized ε-covers along SGD trajectories. Instead of covering the whole hypothesis space, they cover the trajectory-induced class

$$\mathcal{T}_{A,S} = \{w_t(S,\xi) : 0 \le t \le T,\ \xi \in \Xi\}, \tag{32}$$

or its associated predictor class. The resulting complexity is governed by a localized covering number $\log \mathcal{N}\big(\varepsilon, \mathcal{T}_{A,S}, \|\cdot\|\big)$, rather than by a covering number of the entire parameter space. This makes the bound explicitly algorithm dependent. Uniform convergence is no longer imposed on a fixed global class, but on the set of predictors generated by the training dynamics.

Sachs et al. [51] developed this perspective through algorithm-dependent Rademacher complexity. For an algorithm- and data-dependent class $\mathcal{H}_{A,S}$, the empirical complexity can be written as

$$\widehat{\mathfrak{R}}_S(\mathcal{H}_{A,S}) = \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}_{A,S}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(Z_i)\right],$$

where σ_i are Rademacher variables. The important change is not the use of Rademacher complexity itself, which is classical, but the class to which it is applied. The complexity is assigned to a class jointly induced by the algorithm and the sample, not to a fixed hypothesis space chosen before training. This framework also connects earlier results based on fractal dimension, VC-type arguments, and compression-scheme analyses.

Dupuis et al. [156] further lifted this idea to random sets. In their framework, the learning algorithm outputs a data-dependent set $\mathcal{K}_S$, and one studies uniform convergence over this random set:

$$\sup_{h\in\mathcal{K}_S} |L_\mu(h) - L_S(h)|. \tag{33}$$

This provides a high-level PAC-Bayesian perspective on data-dependent uniform bounds and connects localized covers, algorithm-dependent Rademacher complexity, and fractal-dimension-based bounds. It also applies to trajectories of continuous Langevin dynamics and stochastic gradient Langevin dynamics. Thus, the effective object is no longer restricted to a discrete trajectory or a compressed representation; it can be a general data-dependent random set.

The main message of this subsection is that uniform convergence is not abandoned, but relocated. The classical object $\mathcal{H}$ is replaced by $\mathcal{H}_{A,S}$, $\mathcal{T}_{A,S}$, or $\mathcal{K}_S$. Once this replacement is made, capacity control can again become meaningful in overparameterized settings. The later geometric approaches can be viewed as further refinements of the same idea: they describe the effective object not only by covers or Rademacher averages, but also by intrinsic dimension and geometric complexity.

C. Fractal and Geometric Complexity

A related direction describes the objects explored by stochastic optimization through fractal or geometric complexity. The starting observation is that SGD and its variants do not explore the ambient parameter space uniformly. Their trajectories may concentrate on sets with much smaller intrinsic or fractal dimension than the total number of parameters. If such an effective dimension can be controlled, then generalization may depend on this geometric complexity rather than on ambient dimension.

A representative starting point is the work of Şimşekli et al. [157]. Using a continuous-time approximation of stochastic optimization, they viewed SGD trajectories as random paths generated by a Feller process and related generalization to the Hausdorff dimension of the trajectory. Informally, if $\mathcal{T}$ denotes the optimization path, the relevant complexity is no longer d, the ambient parameter dimension, but

$$d_{\text{eff}} = \dim_{\mathrm{H}}(\mathcal{T}). \tag{34}$$

This reframes generalization in terms of the fractal geometry of the path or the invariant structure induced by the optimizer. Their analysis also connects this dimension to the tail behavior of the driving stochastic process, suggesting that heavy-tailed dynamics may correspond to smaller effective capacity.

Camuto et al. [53] extended this idea by modeling stochastic optimization algorithms as random iterated function systems. Instead of focusing only on a single path, they studied the support of the invariant measure induced by the optimizer. A schematic way to express this shift is

$$\text{trajectory geometry} \quad \to \quad \mathrm{supp}(\pi_A),$$

where π_A denotes an invariant measure associated with the stochastic optimizer. The generalization behavior is then linked

to the fractal complexity of this invariant support, which depends on algorithmic factors such as step size, batch size, and problem geometry. This strengthens the interpretation of fractal complexity as an algorithm-induced effective complexity.

Birdal et al. [54] brought this line closer to topological data analysis by introducing persistent homology dimension as a computable proxy for the complexity of optimization trajectories. This suggests that effective geometric complexity can be estimated not only through Hausdorff-type quantities, but also through intrinsic dimension and persistent homology. In this way, the fractal-complexity line moves from abstract geometric characterization toward quantities that can be computed from training trajectories.

Barsbey et al. [55] connected this geometric viewpoint with compression by studying heavy-tailed SGD dynamics and compressibility of overparameterized neural networks. Their results suggest that heavy-tailed optimization may be related not only to smaller fractal dimension, but also to stronger compressibility. This links two apparently different responses to overparameterization: one based on short descriptions, and the other based on low-dimensional geometric structure.

Hodgkinson [56] refined the geometric analysis through lower-tail exponents. Instead of describing complexity only by a global set dimension, this work relates generalization to the local transition behavior of the stochastic optimizer. If $P_t(w, \cdot)$ denotes a local transition kernel, the relevant quantity is tied to the small-ball or lower-tail behavior of this kernel. Thus, effective complexity becomes a property of the optimizer's local dynamics, not only of the final set it visits.

Dupuis [158] further introduced data-dependent fractal dimensions and established fractal-geometry-based generalization bounds over more general fixed or random hypothesis spaces. This removes some reliance on global Lipschitz assumptions in earlier fractal analyses. The work also introduced geometric stability to explain the mutual-information-like terms appearing in the bounds and connected this framework to the topological tools discussed above. Fractal and geometric complexity therefore become part of a broader framework involving random sets, information-type quantities, and computable topological descriptors.

Viewed together, the geometric line reinforces the main conclusion of this section. Once global uniform convergence over the ambient hypothesis space becomes insufficient, one can still retain a capacity-control perspective by measuring the complexity of the smaller objects induced by training: trajectories, invariant measures, local transition structures, or geometric supports. The difference from localized covers and algorithm-dependent Rademacher complexity is mainly the language: the former stays closer to classical learning-theoretic complexity, while the latter uses fractal, intrinsic, and topological notions of effective dimension.

D. Discussion and Limitations

The works discussed in this section are not as unified as stability theory. They are better understood as related responses to the same problem: in modern overparameterized training, the complexity that matters is often not the size of the entire hypothesis space, but the complexity of the effective object explored by the algorithm. Different approaches identify this object in different ways. Compression uses description length; surrogate methods modify the predictor or class on which uniform control is imposed; localized covers and algorithm-dependent Rademacher complexity focus on trajectory- or data-dependent classes; random-set methods provide a more general PAC-Bayesian formulation; and fractal or geometric approaches describe the intrinsic structure of trajectories and invariant supports.

The main contribution of this perspective is therefore a change in the object of analysis. The central question is no longer how large the original class $\mathcal{H}$ is, but how complex the algorithm-induced effective object $\mathcal{H}_{A,S}$ is. This distinguishes the present perspective from the previous three. Stability tracks the propagation of sample perturbations; information theory measures data-output dependence; PAC-Bayes controls prior-posterior complexity; the present line studies the effective capacity of the algorithm-dependent set actually explored during training.

At the same time, this perspective has clear limitations. First, the methodology is not yet standardized. Compression, surrogates, localized covers, random sets, and fractal dimensions share a common motivation, but their technical forms differ substantially. Second, many results rely on problem-specific assumptions, structural modifications, or modeling choices about the geometry of training trajectories. Third, some optimization-dependent generalization results, such as bounds for convex SGD [159], stochastic convex optimization [160], [161], or stochastic minimax optimization [162], also move beyond global capacity control, but do not explicitly rely on effective-class localization. They are therefore better viewed as lying near the boundary of this perspective rather than as its core examples.

For these reasons, this section should not be presented as a fourth algorithmic theory fully parallel to stability. It is more accurately a modern response to the failure of global capacity control in overparameterized learning. Its value lies in showing that capacity-based thinking need not be abandoned: it can be relocated from the global hypothesis space to the compressed model, surrogate class, trajectory class, random set, or geometric support induced by training. In this sense, this perspective complements the previous ones. Stability explains how perturbations propagate, information theory quantifies data dependence, PAC-Bayes controls posterior complexity, and algorithm-dependent complexity asks how the training algorithm itself restricts the set of objects that need to be explained.

VII. Conclusion

Randomized optimization algorithms have become the dominant training paradigm in modern machine learning, and understanding their generalization behavior has therefore become a central problem in learning theory. Unlike classical analyses based mainly on global capacity control, the study of randomized optimization shows that generalization

is deeply tied to the training process itself: how sample perturbations propagate along optimization trajectories, how much information about the training data is retained in the learned model, how posterior complexity is controlled relative to a prior, and how large the effective class explored by the algorithm actually is. In this survey, we reviewed four main perspectives—stability, information-theoretic analysis, PAC-Bayesian analysis, and algorithm-dependent complexity—and showed that they provide complementary views of the same randomized training process rather than isolated and unrelated theories.

At the same time, no single perspective yet offers a complete explanation of generalization in modern stochastic training. Each framework captures an important aspect of the problem, but each also has clear limitations in scope, sharpness, or algorithmic coverage. This suggests that future progress will likely come not from seeking a universal replacement theory, but from strengthening the interaction among these perspectives and extending them to broader and more realistic settings, including non-i.i.d. sampling, distributed and decentralized optimization, modern large-scale training pipelines, and highly overparameterized regimes. We hope this survey provides a coherent roadmap for the field and helps motivate further work toward more unified, algorithm-aware, and practically relevant theories of generalization.

REFERENCES

[1] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[2] M. J. Kearns, R. E. Schapire, and L. M. Sellie, "Toward efficient agnostic learning," in *Proceedings of the annual workshop on Computational learning theory*, p. 341–352, Association for Computing Machinery, 1992.

[3] V. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[4] V. Koltchinskii and D. Panchenko, "Rademacher processes and bounding the risk of function learning," in *High dimensional probability II*, pp. 443–457, Springer, 2000.

[5] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[6] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.

[7] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Learnability, stability and uniform convergence," *Journal of Machine Learning Research*, vol. 11, no. 90, pp. 2635–2670, 2010.

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations*, 2017.

[9] K. Kawaguchi, Y. Bengio, and L. Kaelbling, *Generalization in Deep Learning*. Cambridge University Press, 2022.

[10] S. Chatterjee and P. Zielinski, "On the generalization mystery in deep learning," *arXiv preprint arXiv:2203.10036*, 2022.

[11] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *International Conference on Machine Learning*, vol. 48, pp. 1225–1234, 2016.

[12] I. Kuzborskij and C. Lampert, "Data-dependent stability of stochastic gradient descent," in *International Conference on Machine Learning*, vol. 80, pp. 2815–2824, 2018.

[13] Y. Lei and Y. Ying, "Fine-grained analysis of stability and generalization for stochastic gradient descent," in *International Conference on Machine Learning*, vol. 119, pp. 5809–5819, 2020.

[14] V. Feldman and J. Vondrak, "Generalization bounds for uniformly stable algorithms," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[15] V. Feldman and J. Vondrak, "High probability generalization bounds for uniformly stable algorithms with nearly optimal rate," in *Conference on Learning Theory*, vol. 99, pp. 1270–1279, 2019.

[16] R. Bassily, V. Feldman, C. Guzmán, and K. Talwar, "Stability of stochastic gradient descent on nonsmooth convex losses," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 4381–4391, 2020.

[17] O. Bousquet, Y. Klochkov, and N. Zhivotovskiy, "Sharper bounds for uniformly stable algorithms," in *Conference on Learning Theory*, vol. 125, pp. 610–626, 2020.

[18] Y. Zhou, Y. Liang, and H. Zhang, "Understanding generalization error of SGD in nonconvex optimization," *Machine Learning*, vol. 111, no. 1, pp. 345–375, 2022.

[19] A. Raj, L. Zhu, M. Gurbuzbalaban, and U. Simsekli, "Algorithmic stability of heavy-tailed SGD with general loss functions," in *International Conference on Machine Learning*, vol. 202, pp. 28578–28597, 2023.

[20] W. Mou, L. Wang, X. Zhai, and K. Zheng, "Generalization bounds of SGLD for non-convex learning: Two theoretical viewpoints," in *Conference on Learning Theory*, vol. 75, pp. 605–638, PMLR, 2018.

[21] Y. Klochkov and N. Zhivotovskiy, "Stability and deviation optimal risk bounds with convergence rate $o(1/n)$," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5065–5076, 2021.

[22] A. Banerjee, T. Chen, X. Li, and Y. Zhou, "Stability based generalization bounds for exponential family langevin dynamics," in *International Conference on Machine Learning*, vol. 162, pp. 1412–1449, PMLR, 2022.

[23] T. Sun, D. Li, and B. Wang, "Stability and generalization of decentralized stochastic gradient descent," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 9756–9764, 2021.

[24] X. Deng, L. Shen, S. Li, T. Sun, D. Li, and D. Tao, "Toward understanding the generalizability of delayed stochastic gradient descent," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 9, pp. 7976–7986, 2025.

[25] S. Zeng and Y. Lei, "Stochastic gradient methods: Bias, stability and generalization," *Journal of Machine Learning Research*, vol. 27, no. 6, pp. 1–55, 2026.

[26] D. Russo and J. Zou, "Controlling bias in adaptive data analysis using information theory," in *Artificial Intelligence and Statistics*, pp. 1232–1240, PMLR, 2016.

[27] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[28] A. Pensia, V. Jog, and P.-L. Loh, "Generalization error bounds for noisy, iterative algorithms," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 546–550, IEEE, 2018.

[29] J. Negrea, M. Haghifam, G. K. Dziugaite, A. Khisti, and D. M. Roy, "Information-theoretic generalization bounds for sgld via data-dependent estimates," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] Y. Bu, S. Zou, and V. V. Veeravalli, "Tightening mutual information-based bounds on generalization error," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 121–130, 2020.

[31] M. Haghifam, J. Negrea, A. Khisti, D. M. Roy, and G. K. Dziugaite, "Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9925–9935, 2020.

[32] G. Neu, G. K. Dziugaite, M. Haghifam, and D. M. Roy, "Information-theoretic generalization bounds for stochastic gradient descent," in *Conference on Learning Theory*, pp. 3526–3545, PMLR, 2021.

[33] T. Farghly and P. Rebeschini, "Time-independent generalization bounds for sgld in non-convex settings," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 19836–19846, 2021.

[34] I. Issa, A. R. Esposito, and M. Gastpar, "Generalization error bounds for noisy, iterative algorithms via maximal leakage," in *Conference on Learning Theory*, pp. 4952–4976, PMLR, 2023.

[35] F. Futami and M. Fujisawa, "Time-independent information-theoretic generalization bounds for sgld," *Advances in Neural Information Processing Systems*, vol. 36, pp. 8173–8185, 2023.

[36] H. Wang, Y. Huang, R. Gao, and F. Calmon, "Analyzing the generalization capability of sgld using properties of gaussian channels," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 24222–24234, 2021.

[37] M. Haghifam, B. Rodríguez-Gálvez, R. Thobaben, M. Skoglund, D. M. Roy, and G. K. Dziugaite, "Limitations of information-theoretic generalization bounds for gradient descent methods in stochastic convex optimization," in *International Conference on Algorithmic Learning Theory*, pp. 663–706, PMLR, 2023.

[38] R. Livni, "Information theoretic lower bounds for information theoretic upper bounds," *Advances in Neural Information Processing Systems*, vol. 36, pp. 37716–37727, 2023.

[39] M. Raginsky, A. Rakhlin, and A. Xu, "Information-theoretic stability and generalization," in *Information-Theoretic Methods in Data Science*, pp. 302–329, 2021.

[40] Z. Wang and Y. Mao, "Sample-conditioned hypothesis stability sharpens information-theoretic generalization bounds," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 49513–49541, 2023.

[41] L. T. Dadi and V. Cevher, "Generalization of noisy sgd in unbounded non-convex settings," in *International Conference on Machine Learning*, 2025.

[42] B. London, "A pac-bayesian analysis of randomized learning with application to stochastic gradient descent," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[43] O. Rivasplata, E. Parrado-Hernández, J. S. Shawe-Taylor, S. Sun, and C. Szepesvári, "Pac-bayes bounds for stable algorithms with instance-dependent priors," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[44] E. Parrado-Hernández, A. Ambroladze, J. Shawe-Taylor, and S. Sun, "Pac-bayes bounds with data dependent priors," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3507–3531, 2012.

[45] P. Awasthi, S. Kale, S. Karp, and M. Mohri, "Pac-bayes learning bounds for sample-dependent priors," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 4403–4414, 2020.

[46] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis," in *International Conference on Machine Learning*, pp. 9636–9647, PMLR, 2020.

[47] G. K. Dziugaite and D. Roy, "Entropy-sgd optimizes the prior of a pac-bayes bound: Generalization properties of entropy-sgd and data-dependent priors," in *International Conference on Machine Learning*, pp. 1377–1386, PMLR, 2018.

[48] X. Luo, B. Luo, and J. Li, "Generalization bounds for gradient methods via discrete and continuous prior," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10600–10614, 2022.

[49] E. Clerico, T. Farghly, G. Deligiannidis, B. Guedj, and A. Doucet, "Generalisation under gradient descent via deterministic pac-bayes," in *International Conference on Algorithmic Learning Theory*, 2025.

[50] S. Park, U. Simsekli, and M. A. Erdogdu, "Generalization bounds for stochastic gradient descent via localized ε-covers," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 2790–2802, 2022.

[51] S. Sachs, T. van Erven, L. Hodgkinson, R. Khanna, and U. Şimşekli, "Generalization guarantees via algorithm-dependent rademacher complexity," in *Conference on Learning Theory*, pp. 4863–4880, PMLR, 2023.

[52] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, "Stronger generalization bounds for deep nets via a compression approach," in *International Conference on Machine Learning*, vol. 80, pp. 254–263, 2018.

[53] A. Camuto, G. Deligiannidis, M. A. Erdogdu, M. Gurbuzbalaban, U. Simsekli, and L. Zhu, "Fractal structure and generalization properties of stochastic optimization algorithms," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 18774–18788, 2021.

[54] T. Birdal, A. Lou, L. J. Guibas, and U. Simsekli, "Intrinsic dimension, persistent homology and generalization in neural networks," *Advances in neural information processing systems*, vol. 34, pp. 6776–6789, 2021.

[55] M. Barsbey, M. Sefidgaran, M. A. Erdogdu, G. Richard, and U. Simsekli, "Heavy tails in sgd and compressibility of overparametrized neural networks," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 29364–29378, 2021.

[56] L. Hodgkinson, U. Simsekli, R. Khanna, and M. Mahoney, "Generalization bounds using lower tail exponents in stochastic optimizers," in *International Conference on Machine Learning*, pp. 8774–8795, PMLR, 2022.

[57] B. Dupuis, D. Shariatian, M. Haddouche, A. O. Durmus, and U. Simsekli, "Algorithm-and data-dependent generalization bounds for diffusion models," in *Advances in Neural Information Processing Systems*, 2023.

[58] W. H. Rogers and T. J. Wagner, "A finite sample distribution-free performance bound for local discrimination rules," *The Annals of Statistics*, vol. 6, no. 3, pp. 506–514, 1978.

[59] L. Devroye and T. Wagner, "Distribution-free inequalities for the deleted and holdout error estimates," *IEEE Transactions on Information Theory*, vol. 25, no. 2, pp. 202–207, 1979.

[60] L. Devroye and T. Wagner, "Distribution-free performance bounds with the resubstitution error estimate," *IEEE Transactions on Information Theory*, vol. 25, no. 2, pp. 208–210, 1979.

[61] J. F. Bonnans and A. Shapiro, "Optimization problems with perturbations: A guided tour," *SIAM review*, vol. 40, no. 2, pp. 228–264, 1998.

[62] O. Bousquet and A. Elisseeff, "Algorithmic stability and generalization performance," in *Advances in Neural Information Processing Systems*, p. 178–184, 2000.

[63] M. Kearns and D. Ron, "Algorithmic stability and sanity-check bounds for leave-one-out cross-validation," *Neural Computation*, vol. 11, no. 6, pp. 1427–1453, 1999.

[64] S. Kutin and P. Niyogi, "Almost-everywhere algorithmic stability and generalization error," in *Uncertainty in Artificial Intelligence*, p. 275–282, 2002.

[65] A. Elisseeff, T. Evgeniou, M. Pontil, and L. P. Kaelbing, "Stability of randomized learning algorithms," *Journal of Machine Learning Research*, vol. 6, no. 3, pp. 55–79, 2005.

[66] S. Mukherjee, P. Niyogi, T. Poggio, and R. Rifkin, "Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization," *Advances in Computational Mathematics*, vol. 25, no. 1-3, pp. 161–193, 2006.

[67] A. Maurer, "A second-order look at stability and generalization," in *Conference on Learning Theory*, vol. 65, pp. 1461–1475, 2017.

[68] Z. Deng, H. He, and W. Su, "Toward better generalization bounds with locally elastic stability," in *International Conference on Machine Learning*, vol. 139, pp. 2590–2600, 2021.

[69] X. Yuan and P. Li, "Exponential generalization bounds with near-optimal rates for l_q-stable algorithms," in *International Conference on Learning Representations*, 2023.

[70] X. Yuan and P. Li, "l_2-uniform stability of randomized learning algorithms: Sharper generalization bounds and confidence boosting," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 78580–78592, 2023.

[71] J. Fan and Y. Lei, "High-probability generalization bounds for pointwise uniformly stable algorithms," *Applied and Computational Harmonic Analysis*, vol. 70, p. 101632, 2024.

[72] Y. Lei and Y. Ying, "Sharper generalization bounds for learning with gradient-dominated objective functions," in *International Conference on Learning Representations*, 2021.

[73] Y. Zhang, W. Zhang, S. Bald, V. Pingali, C. Chen, and M. Goswami, "Stability of SGD: Tightness analysis and improved bounds," in *Uncertainty in Artificial Intelligence*, vol. 180, pp. 2364–2373, 2022.

[74] D. Richards and I. Kuzborskij, "Stability & generalisation of gradient descent for shallow neural networks without the neural tangent kernel," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 8609–8621, 2021.

[75] Y. Lei, R. Jin, and Y. Ying, "Stability and generalization analysis of gradient methods for shallow neural networks," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 38557–38570, 2022.

[76] K. Nikolakakis, F. Haddadpour, A. Karbasi, and D. Kalogerias, "Beyond lipschitz: Sharp generalization and excess risk bounds for full-batch GD," in *International Conference on Learning Representations*, 2023.

[77] Y. Lei, T. Sun, and M. Liu, "Minibatch and local SGD: Algorithmic stability and linear speedup in generalization," *Applied and Computational Harmonic Analysis*, vol. 79, p. 101795, 2025.

[78] A. Ramezani-Kebrya, K. Antonakopoulos, V. Cevher, A. Khisti, and B. Liang, "On the generalization of stochastic gradient descent with momentum," *Journal of Machine Learning Research*, vol. 25, no. 22, pp. 1–56, 2024.

[79] X. Pan, X. Li, J. Liu, T. Sun, K. Sun, L. Chen, and Z. Qu, "Stability and generalization for stochastic recursive momentum-based algorithms for (Strongly-)Convex one to k-level stochastic optimizations," in *International Conference on Machine Learning*, vol. 235, pp. 39201–39275, 2024.

[80] X. Pan, J. Liu, H. Kuang, Y. Li, L. Chen, and Z. Qu, "Stability and generalization for stochastic (compositional) optimizations," in *International Joint Conference on Artificial Intelligence*, pp. 6039–6047, 2025.

[81] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *International Conference on Machine Learning*, pp. 681–688, 2011.

[82] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[83] P. Zhou, H. Yan, X. Yuan, J. Feng, and S. Yan, "Towards understanding why lookahead generalizes better than sgd and beyond," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 27290–27304, 2021.

[84] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*, 2020.

[85] X. Zhou and H. Wang, "The generalization error of graph convolutional networks may enlarge with more layers," *Neurocomputing*, vol. 424, pp. 97–106, 2021.

[86] K. Li and Y. Lei, "Generalization and optimization of sgd with lookahead," *arXiv preprint arXiv:2509.15776*, 2025.

[87] C. Tan, J. Zhang, J. Liu, and Y. Gong, "Sharpness-aware lookahead for accelerating convergence and improving generalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[88] C. Tan, J. Zhang, J. Liu, Y. Wang, and Y. Hao, "Stabilizing sharpness-aware minimization through a simple renormalization strategy," *Journal of Machine Learning Research*, vol. 26, no. 68, pp. 1–35, 2025.

[89] M. Schliserman, S. Vansover-Hager, and T. Koren, "Flat minima and generalization: Insights from stochastic convex optimization," *arXiv preprint arXiv:2511.03548*, 2025.

[90] Q. Meng, Y. Wang, W. Chen, T. Wang, Z.-M. Ma, and T.-Y. Liu, "Generalization error bounds for optimization algorithms via stability," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[91] P. Wang, L. Wu, and Y. Lei, "Stability and generalization for randomized coordinate descent," in *International Joint Conference on Artificial Intelligence*, pp. 3104–3110, 2021.

[92] K. Nikolakakis, F. Haddadpour, D. Kalogerias, and A. Karbasi, "Black-box generalization: Stability of zeroth-order learning," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 31525–31541, 2022.

[93] X. Liu, H. Zhang, B. Gu, and H. Chen, "General stability analysis for zeroth-order optimization algorithms," in *International Conference on Learning Representation*, vol. 2024, pp. 19483–19528, 2024.

[94] X. Wu, J. Zhang, and F.-Y. Wang, "Stability-based generalization analysis of distributed learning algorithms for big data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 801–812, 2019.

[95] J. Regatti, G. Tendolkar, Y. Zhou, A. Gupta, and Y. Liang, "Distributed SGD generalizes well under asynchrony," in *Annual Allerton Conference on Communication, Control, and Computing*, pp. 863–870, IEEE, 2019.

[96] D. Richards and P. Rebeschini, "Graph-dependent implicit regularisation for distributed stochastic subgradient descent," *Journal of Machine Learning Research*, vol. 21, no. 34, pp. 1–44, 2020.

[97] X. Deng, T. Sun, S. Li, and D. Li, "Stability-based generalization analysis of the asynchronous decentralized SGD," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 37, pp. 7340–7348, 2023.

[98] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[99] X. Deng, T. Sun, S. Li, D. Li, and X. Lu, "Stability and generalization of asynchronous sgd: Sharper bounds beyond lipschitz and smoothness," in *Advances in Neural Information Processing Systems*, vol. 37, pp. 7675–7713, 2024.

[100] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[101] T. Zhu, F. He, L. Zhang, Z. Niu, M. Song, and D. Tao, "Topology-aware generalization of decentralized SGD," in *International Conference on Machine Learning*, vol. 162, pp. 27479–27503, 2022.

[102] B. Le Bars, A. Bellet, M. Tommasi, K. Scaman, and G. Neglia, "Improved stability and generalization guarantees of the decentralized SGD algorithm," in *International Conference on Machine Learning*, vol. 235, pp. 26215–26240, 2024.

[103] J. Wang and H. Chen, "Towards stability and generalization bounds in decentralized minibatch stochastic gradient descent," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 15511–15519, 2024.

[104] S. Zeng and Y. Lei, "Stability and generalization analysis of decentralized SGD: Sharper bounds beyond lipschitzness and smoothness," in *International Conference on Machine Learning*, vol. 267, pp. 74098–74132, 2025.

[105] H. Ye, T. Sun, and Q. Ling, "Generalization error analysis for attack-free and byzantine-resilient decentralized learning with data heterogeneity," *IEEE Transactions on Signal Processing*, 2026.

[106] X. Hu, Z. Gong, G. Xu, W. Liu, J. Luan, B. Wang, and Y. Liu, "Stability and generalization of zeroth-order decentralized stochastic gradient descent with changing topology," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 17342–17350, 2025.

[107] Q. Li, Y. Liu, M. Zhang, X. Cao, Q. Yin, and L. Shen, "Unveiling the power of multiple gossip steps: A stability-based generalization analysis in decentralized training," in *Advances in Neural Information Processing Systems*, 2025.

[108] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *International Conference on Machine Learning*, pp. 3043–3052, 2018.

[109] P. Wang, Y. Lei, Y. Ying, and D.-X. Zhou, "Stability and generalization for markov chain stochastic gradient methods," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 37735–37748, 2022.

[110] M. Yang, X. Wei, T. Yang, and Y. Ying, "Stability and generalization of stochastic compositional gradient descent algorithms," in *International Conference on Machine Learning*, vol. 235, pp. 56542–56593, 2024.

[111] J. Chen, H. Chen, and B. Gu, "How does black-box impact the learning guarantee of stochastic compositional optimization?," in *Advances in Neural Information Processing Systems*, vol. 37, pp. 107745–107794, 2024.

[112] F. Farnia and A. Ozdaglar, "Train simultaneously, generalize better: Stability of gradient-based minimax learners," in *International Conference on Machine Learning*, pp. 3174–3185, 2021.

[113] Y. Lei, Z. Yang, T. Yang, and Y. Ying, "Stability and generalization of stochastic gradient methods for minimax problems," in *International Conference on Machine Learning*, vol. 139, pp. 6175–6186, 2021.

[114] J. Zhang, M. Hong, M. Wang, and S. Zhang, "Generalization bounds for stochastic saddle point problems," in *International Conference on Artificial Intelligence and Statistics*, pp. 568–576, 2021.

[115] A. Ozdaglar, S. Pattathil, J. Zhang, and K. Zhang, "What is a good metric to study generalization of minimax learners?," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 38190–38203, 2022.

[116] Y. Xing, Q. Song, and G. Cheng, "On the algorithmic stability of adversarial training," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 26523–26535, 2021.

[117] J. Xiao, Y. Fan, R. Sun, J. Wang, and Z.-Q. Luo, "Stability analysis and generalization bounds of adversarial training," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 15446–15459, 2022.

[118] K. Zhang, Y. Wang, and R. Arora, "Stability and generalization of adversarial training for shallow neural networks with smooth activation," in *Advances in Neural Information Processing Systems*, vol. 37, pp. 16160–16193, 2024.

[119] R. Tian and Y. Mao, "Algorithmic stability based generalization bounds for adversarial training," in *International Conference on Learning Representations*, 2025.

[120] Y. Lei, A. Ledent, and M. Kloft, "Sharper generalization bounds for pairwise learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 21236–21246, 2020.

[121] Y. Lei, M. Liu, and Y. Ying, "Generalization guarantee of sgd for pairwise learning," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 21216–21228, 2021.

[122] L. Wu, R. Hu, and Y. Lei, "Stability-based generalization analysis of randomized coordinate descent for pairwise learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 21545–21553, 2025.

[123] J. Wang, J. Chen, H. Chen, B. Gu, W. Li, and X. Tang, "Stability-based generalization analysis for mixtures of pointwise and pairwise learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 10113–10121, 2023.

[124] J. Chen, H. Chen, X. Jiang, B. Gu, W. Li, T. Gong, and F. Zheng, "On the stability and generalization of triplet learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 7033–7041, 2023.

[125] A. Maurer, "Algorithmic stability and meta-learning," *Journal of Machine Learning Research*, vol. 6, no. 33, pp. 967–994, 2005.

[126] M. Al-Shedivat, L. Li, E. Xing, and A. Talwalkar, "On data efficiency of meta-learning," in *International Conference on Artificial Intelligence and Statistics*, vol. 130, pp. 1369–1377, 2021.

[127] A. Farid and A. Majumdar, "Generalization bounds for meta-learning via PAC-Bayes and uniform stability," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 2173–2186, 2021.

[128] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 5469–5480, 2021.

[129] J. Guan, Y. Liu, and Z. Lu, "Fine-grained analysis of stability and generalization for modern meta learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 18487–18500, 2022.

[130] Y. Wang and R. Arora, "On the stability and generalization of meta-learning," in *Advances in Neural Information Processing Systems*, vol. 37, pp. 83665–83710, 2024.

[131] W. Ding, J. Liu, L. Chen, X. Su, T. Sun, F. Wu, and Z. Qu, "On the stability and generalization of meta-learning: the impact of inner-levels," in *Advances in Neural Information Processing Systems*, 2025.

[132] J. Chen, H. Chen, B. Gu, and H. Deng, "Fine-grained theoretical analysis of federated zeroth-order optimization," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 54496–54508, 2023.

[133] Y. Liu, Q. Li, J. Tan, Y. Shi, L. Shen, and X. Cao, "Understanding the stability-based generalization of personalized federated learning," in *International Conference on Learning Representations*, 2025.

[134] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[135] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[136] T. Zhang, "Information-theoretic upper and lower bounds for statistical estimation," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1307–1321, 2006.

[137] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro, "Exploring generalization in deep learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[138] A. T. Lopez and V. Jog, "Generalization error bounds using wasserstein distances," in *ieee information theory workshop (ITW)*, pp. 1–5, IEEE, 2018.

[139] J. Zhang, T. Liu, and D. Tao, "Going deeper, generalizing better: An information-theoretic view for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 16683–16695, 2023.

[140] Y. Dong, T. Gong, H. Chen, and C. Li, "Understanding the generalization ability of deep learning algorithms: a kernelized rényi's entropy perspective," *arXiv preprint arXiv:2305.01143*, 2023.

[141] M. Haghifam, S. Moran, D. M. Roy, and G. K. Dziugiate, "Understanding generalization via leave-one-out conditional mutual information," in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2487–2492, IEEE, 2022.

[142] D. A. McAllester, "Some PAC-Bayesian theorems," in *Conference on Computational Learning Theory*, pp. 230–234, 1998.

[143] D. A. McAllester, "PAC-Bayesian model averaging," in *Conference on Computational Learning Theory*, pp. 164–170, Association for Computing Machinery, 1999.

[144] M. Seeger, "Pac-bayesian generalisation error bounds for gaussian process classification," *Journal of Machine Learning Research*, vol. 3, pp. 233–269, 2002.

[145] O. Catoni, "Pac-bayesian supervised classification: the thermodynamics of statistical learning," *arXiv preprint arXiv:0712.0248*, 2007.

[146] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," *arXiv preprint arXiv:1703.11008*, 2017.

[147] A. Lacasse, F. Laviolette, M. Marchand, P. Germain, and N. Usunier, "Pac-bayes bounds for the risk of the majority vote and the variance of the gibbs classifier," in *Advances in Neural information processing systems*, vol. 19, 2006.

[148] P. Germain, A. Lacasse, F. Laviolette, M. Marchand, and J.-F. Roy, "Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm," *arXiv preprint arXiv:1503.08329*, 2015.

[149] V. Zantedeschi, P. Viallard, E. Morvant, R. Emonet, A. Habrard, P. Germain, and B. Guedj, "Learning stochastic majority votes by minimizing a pac-bayes generalization bound," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 455–467, 2021.

[150] G. K. Dziugaite and D. M. Roy, "Data-dependent pac-bayes priors via differential privacy," in *Advances in neural information processing systems*, vol. 31, 2018.

[151] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-sgd: Biasing gradient descent into wide valleys," *arXiv preprint arXiv:1611.01838*, 2016.

[152] V. Nagarajan and J. Z. Kolter, "Uniform convergence may be unable to explain generalization in deep learning," in *Advances in neural information processing systems*, vol. 32, 2019.

[153] D. J. Foster, A. Sekhari, and K. Sridharan, "Uniform convergence of gradients for non-convex learning and optimization," in *Advances in neural information processing systems*, vol. 31, 2018.

[154] J. Negrea, G. K. Dziugaite, and D. Roy, "In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors," in *International Conference on Machine Learning*, pp. 7263–7272, 2020.

[155] M. Gastpar, I. Nachum, J. Shafer, and T. Weinberger, "Fantastic generalization measures are nowhere to be found," *arXiv preprint arXiv:2309.13658*, 2023.

[156] B. Dupuis, P. Viallard, G. Deligiannidis, and U. Simsekli, "Uniform generalization bounds on data-dependent hypothesis sets via pac-bayesian theory on random sets," *Journal of Machine Learning Research*, vol. 25, no. 409, pp. 1–55, 2024.

[157] U. Simsekli, O. Sener, G. Deligiannidis, and M. A. Erdogdu, "Hausdorff dimension, heavy tails, and generalization in neural networks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 5138–5151, 2020.

[158] B. Dupuis, G. Deligiannidis, and U. Simsekli, "Generalization bounds using data-dependent fractal dimensions," in *International conference on machine learning*, pp. 8922–8968, PMLR, 2023.

[159] J. Hendrickx and A. Olshevsky, "Convex sgd: Generalization without early stopping," *arXiv preprint arXiv:2401.04067*, 2024.

[160] V. Feldman, "Generalization of erm in stochastic convex optimization: The dimension strikes back," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[161] M. Schliserman, U. Sherman, and T. Koren, "The dimension strikes back with gradients: Generalization of gradient methods in stochastic convex optimization," *arXiv preprint arXiv:2401.12058*, 2024.

[162] B. Zhu, S. Li, and Y. Liu, "Towards sharper risk bounds for minimax problems," *arXiv preprint arXiv:2410.08497*, 2024.

Jiahuan Wang received the the M.S. degree in mathematics from Huazhong Agricultural University, Wuhan, China, in 2024. He is currently pursuing the Ph.D. degree at the College of Computer Science and Technology, National University of Defense Technology. His research interests include stochastic optimization, machine learning theory, and deep learning.

Xiaoge Deng received the BS degree in mathematics from the University of Science and Technology of China (USTC), Hefei, China in 2018. He received the Ph.D. degree at the School of Computer, National University of Defense Technology (NUDT) in 2024. His research interests include optimization for machine learning and distributed systems.

Ziqing Wen received the BS degree in mathematics from SouthWest Jiao Tong University in 2022. He is currently pursuing the Phd degree at the School of Computer National University of Defense Technology. His research interests include optimization for machine learning and generative AI.

Ping Luo is currently pursuing the Ph.D. degree in Computer Science and Technology from National University of Defense Technology (NUDT), Changsha, China. He received his MA.Eng. degree in Computer Science and Technology with Hainan University, Haikou, China. His research interests include Convex Optimization, Artificial Intelligence.

Dongsheng Li received the BSc (Hons.) and PhD (Hons.) degrees in computer science from the National University of Defense Technology, Changsha, China in 1999 and 2005, respectively. He is currently a full professor at the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology. His research interests include parallel and distributed computing, cloud computing, and large-scale data management. He was awarded the prize of the National Excellent Doctoral Dissertation of China by the Ministry of Education of China in 2008. His research interests include distributed systems, cloud computing, and Big Data processing.

Tao Sun received his Ph.D. degree from National University of Defense Technology in 2018. Currently, he is an Associate Professor with the College of Computer Science and Technology, National University of Defense Technology. His research interests include optimization for machine learning, reinforcement learning, distributed systems, and neural networks.

Xinwang Liu received the Ph.D. degree from National University of Defense Technology (NUDT), Changsha, China. He is now Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. He has published 300+ peerreviewed papers, including IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, ICML, NeurIPS, CVPR, etc. He is on the editorial board of TKDE, TNNLS, and TCYB. He also serves/served as Area Chair for CCF-A/B ranked international conferences for more than 30 times.