



PISP: Projected-Space Inference of Stellar Parameters

Jun-Chao Liang^{1,2}, Yin-Bi Li¹, A-Li Luo^{1,2,3}, Shuo Li^{1,2}, Xiao-Xiao Ma^{1,2}, Hai-Ling Lu^{1,2}, Shu-Guo Ma¹, Ming-Hui Jia^{4,5}, Shuo Ye^{1,2}, Hao Zeng^{1,2}, Ke-Fei Wu¹, Zhi-Hua Zhong⁶, Xiao Kong¹, Li-Li Wang⁷, and

Hugh R. A. Jones⁸

¹ National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, People's Republic of China; ybli@bao.ac.cn, lal@nao.cas.cn

² School of Astronomy and Space Science, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

³ University of Chinese Academy of Sciences, Nanjing 211135, People's Republic of China

⁴ Institute of Automation, Chinese Academy of Sciences, Beijing 100190, People's Republic of China

⁵ School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

⁶ Center for Spatial Information Science, The University of Tokyo, Tokyo 153-8505, Japan

⁷ School of Computer and Information, Dezhou University, Dezhou 253023, People's Republic of China

⁸ School of Physics, Astronomy and Mathematics, University of Hertfordshire, UK

Received 2025 September 21; revised 2026 March 6; accepted 2026 March 6; published 2026 April 16

Abstract

To improve the accuracy and efficiency of high-dimensional stellar parameter inference in large spectroscopic datasets, we propose a projection-assisted parameter-inference framework—Projected-Space Inference of Stellar Parameters (PISP). PISP constructs an orthonormal basis and optimizes in the projected space, reducing the impact of parameter correlations on inference. The basis is constructed using either principal component analysis (PCA) or the active-subspace (AS) method and is combined with two inference strategies—Non-L1, which optimizes the projection coefficients for a user-specified projected dimensionality, and L1, which introduces L1 regularization in the full projected space to adaptively select projection directions—yielding four strategies: PCA-Non-L1, AS-Non-L1, PCA-L1, and AS-L1. For different computational scenarios, we implement two versions: PISP-CurveFit for fast single-spectrum inference and PISP-Adam for large-scale GPU-parallel inference. Using a fully connected neural network and a residual network as spectral emulators, we evaluate PISP on Kurucz synthetic spectra and on 722,896 APOGEE DR17 observed spectra. Compared to the baseline strategy, PISP improves inference accuracy for multiple parameters across all emulator–optimizer combinations. In synthetic data, PCA-L1 performs best, reducing the standard deviation of differences ($\sigma(\Delta)$) by at least 0.01 dex for 12 of 20 elemental abundances, with [N/H], [O/H], [Na/H], [Co/H], [P/H], [V/H], [Cu/H] showing 0.05–0.72 dex reductions. In observed data, PCA-Non-L1 reduces $\sigma(\Delta)$ by >30 K for effective temperature and by at least 0.01 dex for 9 of 17 elemental abundances, with [O/H], [Na/H], [V/H] showing 0.05–0.20 dex reductions, while achieving a $\sim 4\times$ efficiency gain, slightly outperforming PCA-L1.

Unified Astronomy Thesaurus concepts: [Astronomy data analysis \(1858\)](#); [Chemical abundances \(224\)](#); [Fundamental parameters of stars \(555\)](#); [GPU computing \(1969\)](#); [Surveys \(1671\)](#)

1. Introduction

In recent years, large spectroscopic surveys such as the Radial Velocity Experiment (M. Steinmetz et al. 2006), the Sloan Digital Sky Survey (SDSS; B. Yanny et al. 2009; S. R. Majewski et al. 2017; A. Almeida et al. 2023), the Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST; X.-Q. Cui et al. 2012; G. Zhao et al. 2012; A.-L. Luo et al. 2015), the Galactic Archaeology with HERMES (GALAH; G. M. De Silva et al. 2015), the Gaia Radial Velocity Spectrometer (RVS; Gaia Collaboration et al. 2023; A. Recio-Blanco et al. 2023), and the Dark Energy Spectroscopic Instrument (DESI; DESI Collaboration et al. 2026), among others, have provided an unprecedented observational foundation for studies of stellar physics and Galactic structure.

These datasets enable the inference of key stellar parameters, such as effective temperature (T_{eff}), surface gravity ($\log g$), and the abundances of dozens of elements. These stellar parameters constitute fundamental data of significant importance for many areas of contemporary astronomy and

astrophysics, ranging from stellar evolution and the formation of planetary systems to the evolution of the Milky Way and external galaxies, as well as cosmology (P. E. Nissen & B. Gustafsson 2018). To achieve efficient and robust parameter inference, various surveys have developed parameter-processing pipelines, such as the APOGEE Stellar Parameter and Chemical Abundances Pipeline (ASPCAP; A. E. García Pérez et al. 2016), the data-driven approach to stellar-label determination used in GALAH DR1/DR2 (The Cannon; M. Ness et al. 2015; S. L. Martell et al. 2017; S. Buder et al. 2018), the Spectroscopy Made Easy adopted in GALAH DR3 (SME; J. A. Valenti & N. Piskunov 1996; N. Piskunov & J. A. Valenti 2017; S. Buder et al. 2021), the self-consistent ab initio fitting of stellar spectra used in GALAH DR4 (The Payne; Y.-S. Ting et al. 2019; S. Buder et al. 2025), and the General Stellar Parametriser-spectroscopy (GSP-Spec) used for Gaia DR3 RVS (A. Recio-Blanco et al. 2023). These pipelines implement the mapping from the stellar-parameter space to the spectral-flux space (spectral emulator), either by invoking physical spectral-synthesis models or by constructing data-driven regression models, and they combine such models with optimization methods appropriate to their implementations to iteratively search for the optimal parameter



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Table 1
Parameter Ranges of the Kurucz Synthetic Spectral Grid

Parameter	Minimum	Maximum
T_{eff} (K)	3007	7997
$\log g$ (dex)	0.0	5.3
v_{turb} (km s ⁻¹)	0.0	3.0
v_{macro} (km s ⁻¹)	0.1	30.0
[C/H] (dex)	-2.5	0.9
[N/H] (dex)	-2.5	1.7
[O/H] (dex)	-2.5	1.6
[Na/H] (dex)	-2.5	1.7
[Mg/H] (dex)	-2.5	1.7
[Al/H] (dex)	-2.5	1.7
[Si/H] (dex)	-2.6	0.8
[P/H] (dex)	-2.5	1.7
[S/H] (dex)	-2.5	1.7
[K/H] (dex)	-2.5	1.4
[Ca/H] (dex)	-2.5	1.8
[Ti/H] (dex)	-2.5	1.8
[V/H] (dex)	-2.5	1.7
[Cr/H] (dex)	-2.5	1.7
[Mn/H] (dex)	-2.5	1.7
[Fe/H] (dex)	-1.5	0.5
[Co/H] (dex)	-2.5	1.7
[Ni/H] (dex)	-2.5	1.8
[Cu/H] (dex)	-2.5	1.7
[Ge/H] (dex)	-2.4	1.2
¹² C/ ¹³ C	0.0	100.0

combination that minimizes the discrepancy between the spectrum predicted by the emulator and the observed spectrum.

In practical applications of large-scale spectroscopic parameterization, stellar parameters often exhibit significant correlations, arising from physical couplings in stellar structure and evolution and also reflected in the statistical properties of the observations. Recent studies have further revealed such structural characteristics of the stellar-parameter space in the Galactic disk: elemental abundances display complex correlation patterns (A. R. Casey et al. 2019; D. H. Weinberg et al. 2019, 2022; E. J. Griffith et al. 2022, 2025; M. K. Ness et al. 2022; Y.-S. Ting & D. H. Weinberg 2022), and they manifest as rigidly coupled relations in samples of F- and G-type dwarfs, namely that multiple elemental abundances (e.g., [O/H], [Si/H], [Ca/H]) can be expressed linearly by T_{eff} , $\log g$, [Fe/H], and [Mg/H] (J. Mead et al. 2025). However, existing parameter-inference methods and optimization algorithms typically solve directly in the original parameter space without explicit treatment of such correlations. In the tests made for this study, we observe that when strong correlations exist among parameters, the optimization process more readily exhibits reduced convergence speed and degraded inference accuracy, suggesting that parameter correlations may affect inference performance to some extent.

In the field of high-dimensional optimization, researchers have proposed various strategies to improve efficiency and stability, such as linear embeddings (Z. Wang et al. 2016; B. Letham et al. 2020; E. Raponi et al. 2020; Y. Mei & H. Wang 2021; L. Pretsch et al. 2024) and nonlinear embeddings (A. Tripp et al. 2020; K. Antonov et al. 2022; N. Maus et al. 2022; O. Boyar & I. Takeuchi 2024; J. Chu et al. 2024). These methods are mainly applied within optimization frameworks such as Bayesian optimization and

evolutionary algorithms. Their core idea is to reduce the effective optimization dimensionality by searching within an embedded space, thereby mitigating, to some extent, high-dimensional optimization challenges arising from parameter correlations or redundant dimensions.

To alleviate the impact of correlations among high-dimensional stellar parameters on inference accuracy and convergence efficiency, and inspired by embedding ideas in high-dimensional optimization, we propose the Projected-Space Inference of Stellar Parameters (PISP): during the inference stage, an orthonormal projection basis is introduced to map stellar parameters to projection coefficients and optimization is then carried out in the coefficient space. The orthonormal basis can be constructed using either principal component analysis (PCA; H. Hotelling 1933; H. Abdi & L. J. Williams 2010) or the active-subspace (AS) method (P. G. Constantine et al. 2014; K. Musayeva & M. Binois 2024). The PCA basis is obtained from the covariance matrix of stellar parameters in the training set, aiming to decorrelate parameters in a statistical sense. In contrast, the AS basis is derived from gradients of the spectral emulator evaluated on the training set. It is not constructed to decorrelate parameters but instead identifies the directions in parameter space to which the spectrum is most sensitive and concentrates the optimization on these key directions, thereby partially alleviating the adverse impact of parameter correlations on optimization convergence and inference stability.

The remainder of this paper is organized as follows. Section 2 describes the data used in this work. Section 3 presents the spectral emulators and the PISP methodology. Section 4 determines the hyperparameters of PISP and evaluates its inference performance. Finally, Section 5 concludes the work.

2. Data

PISP performs stellar-parameter inference using spectral emulators trained on Kurucz synthetic spectra. To assess its performance, we apply PISP to both Kurucz synthetic spectra and APOGEE DR17 observed spectra.

2.1. Synthetic Spectra from Kurucz Models

The synthetic spectra used in this study were provided by Y.-S. Ting. This dataset was generated following the same procedure as the refined synthetic model grid described by Y.-S. Ting et al. (2019), but at a larger scale, comprising 14,623 synthetic spectra that cover a 25-dimensional stellar-parameter space (Table 1); each spectrum consists of 7214 flux pixels. We divided the dataset into a training set (12,000 spectra), a validation set (1312 spectra), and a test set (1311 spectra), for training the spectral emulator and evaluating both its flux-reconstruction error and the performance of PISP on synthetic spectra.

We emphasize that this dataset is not a regular Cartesian grid in the 25-dimensional stellar-parameter space but was instead constructed using the two-stage sampling strategy described in Y.-S. Ting et al. (2019, Section 2.3). This strategy concentrates the synthetic spectral calculations in the local neighborhood of stellar-parameter space that is actually populated by real stars, thereby avoiding the exponential growth in model numbers inherent to a uniformly sampled high-dimensional grid and enabling stable spectral interpolation and parameter inference within this effective parameter domain.

The models were computed with ATLAS12 and SYNTHE under a one-dimensional local thermodynamic equilibrium framework. The atmospheric structure is divided into 80 Rosseland optical-depth layers, with a maximum Rosseland depth of $\tau_{R,\max} = 1000$. Numerical convergence in each atmospheric layer is automatically verified. The reference abundances adopt the solar composition of M. Asplund et al. (2009) and the stellar parameters of Arcturus from I. Ramírez & C. Allende Prieto (2011), which employs an improved H-band atomic line list whose strong-line parameters are calibrated against solar and Arcturus observations. After convergence of the model atmospheres, synthetic spectra are first computed at a nominal resolution of $R = 300,000$, then degraded to APOGEE’s observational resolution by convolution with APOGEE’s average line-spread function, and finally normalized following M. Ness et al. (2015).

2.2. APOGEE Survey Data

APOGEE is one of the programs in SDSS-III (S. R. Majewski et al. 2017) and SDSS-IV (M. R. Blanton et al. 2017). It employs a near-infrared spectrograph with a resolving power of $R \sim 22,500$ over a wavelength range of 1.51–1.70 μm . The primary targets are red giants in the Milky Way, with additional red giants in the Large Magellanic Cloud and other nearby dwarf galaxies, as well as a large number of FGKM dwarfs (V. V. Smith et al. 2021).

In the 17th data release (DR17), APOGEE includes 657,135 unique targets and approximately 2.6 million individual visit spectra, and provides visit-combined spectra produced using pixel-based weighting and a more global weighting.⁹ DR17 reports atmospheric parameters, radial velocities, and chemical abundances for more than 20 elements (Abdurro’uf et al. 2022).

In this work, using the official parameter catalog as the reference, we infer atmospheric parameters and elemental abundances for 722,896¹⁰ visit-combined spectra. The inferred parameters are compared with the official catalog to evaluate the reliability of PISP on observed data.

2.3. Data Preprocessing

Before using PISP to infer high-dimensional parameters from APOGEE stellar spectra, we perform the following preprocessing steps:

1. *Spectral normalization.* To reduce shape differences between APOGEE and synthetic spectra, we normalize the APOGEE spectra prior to inference following M. Ness et al. (2015). This method, based on the data-driven model The Cannon, selects wavelength pixels with minimal response to stellar labels, fits a fourth-order polynomial separately in each of APOGEE’s three spectral bands to approximate the pseudocontinuum, and finally divides the flux by this pseudocontinuum to obtain the normalized spectrum.
2. *Masking outlier pixels.* To ensure reliable spectral fitting, we adopt a dual-mask strategy to exclude abnormal

pixels. First, we directly adopt the mask table from Y.-S. Ting et al. (2019), constructed by comparing synthetic spectra of the Sun and Arcturus with APOGEE observations, which flags pixels with differences between the normalized synthetic and observed spectra exceeding 2%. In parallel, we apply the official APOGEE mask to remove instrument- and observation-related bad pixels. Taking the union of the two yields a per-pixel binary mask $A_{\lambda_i} \in \{0, 1\}$, where $A_{\lambda_i} = 0$ denotes a bad pixel, $A_{\lambda_i} = 1$ denotes a valid pixel, and λ_i denotes the i th wavelength pixel.

3. Methodology

This section first introduces the spectral emulators used for parameter inference (Section 3.1), then details the steps of PISP (Section 3.2), and finally presents a baseline strategy—direct optimization in the standardized stellar-parameter space—to evaluate the effectiveness of PISP (Section 3.3).

3.1. Spectral Emulator

To assess the behavior of PISP with different types of spectral emulators, we construct two architectures: a fully connected neural network (FNN)-based emulator and a residual network (ResNet)-based emulator.

1. *FNN emulator.* The input is a 25-dimensional stellar-label vector and the output is a 7214-dimensional APOGEE flux vector. The network has three hidden layers (with 256, 512, and 1024 units, respectively); all hidden layers use ReLU activations and the output layer is linear.
2. *ResNet emulator.* The input is a 25-dimensional stellar-label vector and the output is a 7214-dimensional APOGEE flux vector. The architecture comprises three fully connected layers and seven deconvolution layers, with residual connections in layers two to six. This implementation follows an open-source reference¹¹ and is used to verify the applicability of PISP across network architectures.

Both emulators share the same training strategy. First, each of the 25 input parameters is scaled independently to the interval $[-0.5, 0.5]$ to alleviate numerical instabilities caused by differences in parameter scales (J. Nocedal & S. J. Wright 2006). Specifically,

$$\theta = \frac{\theta_{\text{raw}} - \min(\theta_{\text{raw}}^{\text{train}})}{\max(\theta_{\text{raw}}^{\text{train}}) - \min(\theta_{\text{raw}}^{\text{train}})} - 0.5, \quad (1)$$

where θ_{raw} denotes the raw stellar parameters, $\max(\theta_{\text{raw}}^{\text{train}})$ and $\min(\theta_{\text{raw}}^{\text{train}})$ are the per-dimension maxima and minima computed on the training set, respectively, and θ is the normalized parameter vector. We use the mean absolute error (MAE) as the loss function; all models are trained for up to 10^5 epochs with Adam for parameter updates. In addition, we employ a plateau learning-rate scheduler (ReduceLROnPlateau)¹² based on the validation-set MAE. If the validation

⁹ https://www.sdss4.org/dr17/irspec/spectral_combination

¹⁰ The official catalog contains 733,901 entries; we exclude 10,996 empty FITS entries and nine spectra that failed normalization, yielding 722,896 visit-combined spectra.

¹¹ https://github.com/tingyuansen/The_Payne/blob/master/The_Payne/training.py

¹² https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

MAE does not improve for 500 consecutive epochs, the learning rate is reduced by a factor of 0.9, with a cooldown of 200 epochs, down to a minimum of 10^{-8} . If, after reaching the minimum learning rate, the validation MAE still fails to improve for 5000 consecutive epochs, training is stopped early to avoid overfitting. Finally, we save the weights corresponding to the lowest validation MAE as the final configuration for each emulator, which are then used for subsequent PISP inference optimization. This unified training framework ensures comparability between the two emulators and provides a basis for evaluating the effectiveness of PISP.

3.2. PISP

Given the widespread use of CurveFit (`scipy.optimize.curve_fit`; P. Virtanen et al. 2020) and Adam (D. P. Kingma & J. Ba 2014) in stellar-parameter inference (M. Ness et al. 2015; Y.-S. Ting et al. 2019; M. Xiang et al. 2019; J. Imig et al. 2022; X. Zhang et al. 2023; T. Róžański et al. 2025; J.-C. Liang et al. 2026), we implement two algorithmic variants to accommodate different computational scenarios: PISP-CurveFit and PISP-Adam. PISP-CurveFit targets CPU environments and is suitable for fast inversion of individual spectra or small-scale tasks; PISP-Adam is mainly implemented based on PyTorch (A. Paszke et al. 2019), targets GPU environments, and supports large-scale parallelism and batch processing to improve throughput and convergence efficiency. Each variant includes four inference strategies: PCA-Non-L1 and AS-Non-L1 optimize the projection coefficients under a user-specified projected dimensionality; while PCA-L1 and AS-L1 introduce L1 regularization in the full projected space to enable adaptive selection of projection directions. PISP is invoked only at the inference stage and can act directly on pretrained spectral emulators without additional training; while improving inference accuracy, it effectively reduces computational cost.

3.2.1. PISP-CurveFit

PISP-CurveFit is mainly used for fast inference of stellar parameters from single spectra and can be combined with the Joblib library (G. Varoquaux et al. 2024) for large survey datasets. The method consists of the following steps:

1. *Single-spectrum storage.* To support multicore parallel processing, each normalized observed spectrum from Section 2.3 and its corresponding official pixel mask are stored together in a single `.npz` file. This format allows fast loading via the NumPy library (C. R. Harris et al. 2020), improving input/output throughput and parallel performance.
2. *Reconstruct standardized stellar parameters.* PISP provides two orthonormal-basis constructions for reconstructing the standardized stellar-parameter vector, based on PCA and the AS method. The optimal construction can be selected for a given target dataset (the basis-selection strategy is described in Section 4.2.1). To present both within a unified framework, we write the reconstruction in the standardized parameter space as

$$\theta_i = w_{i,1:n} \cdot z_{1:n}^T + \mu_\theta. \quad (2)$$

Here, θ_i is the standardized stellar-parameter vector (1×25) of the i th spectrum in an arbitrary dataset; z is the basis matrix (25×25); $z_{1:n}$ denotes the first n

columns of z ($25 \times n$), and thus $z_{1:n}^T$ has dimension $n \times 25$; $w_{i,1:n}$ is the coefficient vector to be optimized ($1 \times n$); and μ_θ is the reference center vector (1×25). For PCA, μ_θ is the mean vector of the standardized training-set parameters (denoted by θ^{tr}) along each dimension; for AS, $\mu_\theta = 0$. We treat n as a hyperparameter that controls the projected dimensionality: $n < 25$ corresponds to reconstruction in a lower-dimensional subspace, whereas $n = 25$ corresponds to an orthogonal reparameterization without dimensionality reduction. The choice and discussion of n are provided in Section 4.2.1. The basis matrix z is precomputed once on the training set and is then fixed. For PCA, z is formed by the eigenvectors of the sample covariance matrix of $\theta^{\text{tr}} - \mu_\theta$, ordered by descending eigenvalues; we implement this procedure using `sklearn.decomposition.PCA` (F. Pedregosa et al. 2011). For AS, we treat the spectral emulator $g(\cdot)$ as a vector-valued function and adopt the shared active subspace for vector-valued functions proposed by K. Musayeva & M. Binois (2024, Section 3.1.2) to construct z . Specifically, for the standardized parameters of the k th training sample, θ_k^{tr} , we define the pixel-averaged sensitivity vector

$$s_k = \frac{1}{m_1} \sum_{j=1}^{m_1} \frac{\partial g(\theta)_j}{\partial \theta} \Big|_{\theta=\theta_k^{\text{tr}}},$$

where m_1 is the flux dimensionality ($m_1 = 7214$), j is the wavelength index, and s_k is a row vector of dimension 1×25 . The partial derivatives are computed via `torch.func.jacrev`. The pixel-averaged sensitivity vector reflects the global spectral sensitivity across all wavelength pixels, where contributions from near-zero gradient pixels and cancellation between pixels with gradients of opposite signs are naturally incorporated, rather than localized sensitivity at individual lines. This averaging step trades off local detail for global robustness, which may reduce the influence of parameters that affect only a few pixels. We further construct

$$C = \frac{1}{m_2} \sum_{k=1}^{m_2} s_k^T s_k,$$

where m_2 is the number of training samples ($m_2 = 12,000$). This summation provides an empirical estimate of the expected outer product of the pixel-averaged sensitivity vector with itself over the training set. Finally, we perform an eigen-decomposition of C using `torch.linalg.eigh`, order the eigenvectors by descending eigenvalues, and use them to form the basis matrix z .

3. *Define the objective function.* For the i th spectrum F_i (1×7214) and its binary pixel-mask vector A_i (1×7214), we define the objective function in the standardized parameter space as

$$\mathcal{L}_{\text{CurveFit}}^{\text{raw}} = \sum_j (F_{i,j} \cdot A_{i,j} - g(\theta)_j \cdot A_{i,j})^2, \quad (3)$$

where j indexes wavelength pixels and $g(\cdot)$ denotes the spectral emulator. Substituting Equation (2) into Equation (3) and adding an L1 regularization term yields the objective function with respect to the basis

coefficients

$$\mathcal{L}_{\text{CurveFit}}^{\text{PISP}} = \sum_j (F_{i,j} \cdot A_{i,j} - g(w_{i,1:n} \cdot z_{1:n}^T + \mu_\theta)_j \cdot A_{i,j})^2 + \alpha \cdot \sum_t |w_{i,t}|, \quad (4)$$

where $\alpha \geq 0$ is the regularization coefficient, and $w_{i,t}$ is the t th basis coefficient ($t = 1, \dots, n$). If $\alpha = 0$, the method corresponds to the PCA-Non-L1 and AS-Non-L1 strategies, in which only the first n coefficients $w_{i,1:n}$ are optimized (n is a hyperparameter, determined in Section 4.2.1). If $\alpha > 0$, the method corresponds to the PCA-L1 and AS-L1 strategies, for which we set $n = 25$ and optimize all coefficients $w_{i,1:25}$ (α is the hyperparameter of this strategy, also determined in Section 4.2.1). The PCA-Non-L1 and AS-Non-L1 strategies perform truncation, discarding the $25-n$ low-variance principal components or the $25-n$ inactive directions, whereas the PCA-L1 and AS-L1 strategies induce sparsity across the full set of 25 basis directions via regularization. The purpose of designing these four strategies is to evaluate how the basis choice and constraint affect inference accuracy, and to adapt to different datasets.

4. *Minimization of the objective function.* For the i th spectrum F_i , PISP-CurveFit provides four inference strategies: PCA-Non-L1 ($\alpha = 0$), PCA-L1 ($\alpha > 0$), AS-Non-L1 ($\alpha = 0$), and AS-L1 ($\alpha > 0$). All four strategies iteratively solve for the minimizer of the objective function in Equation (4) using the trust region reflective (trf) algorithm (T. F. Coleman & Y. Li 1996; M. A. Branch et al. 1999) via CurveFit. The initial values are uniformly set to the basis coefficients computed via Equation (2) from the standardized stellar parameters corresponding to the nearest-neighbor spectrum of F_i in the training set.¹³ Following Y.-S. Ting et al. (2019), we set the convergence tolerances (ftol and xtol) to 5×10^{-4} . Upon convergence, the optimal basis coefficients $w_{i,1:n}^*$ are substituted into Equation (2) to obtain the standardized stellar parameters, which are then transformed back to physical units via the inverse transformation of Equation (1) to yield the final stellar parameters.

5. *Estimating parameter errors.* We define the residual row vector function for the i th spectrum F_i as

$$r(w_{i,1:n}) = \begin{cases} \Delta f_i, & \text{(a)} \\ \begin{bmatrix} (\Delta f_i)^T \\ -\sqrt{\alpha}|w_{i,1}| \\ \vdots \\ -\sqrt{\alpha}|w_{i,n}| \end{bmatrix}, & \text{(b)} \end{cases}, \quad (5)$$

where (a) denotes PCA-Non-L1 and AS-Non-L1, and (b) denotes PCA-L1 and AS-L1. Here, Δf_i is the row vector formed by extracting elements with mask $A_{i,j} = 1$ from $F_i - g(w_{i,1:n} \cdot z_{1:n}^T + \mu_\theta)$ in wavelength order ($1 \times d$, where $d = \sum_j A_{i,j}$). Once the optimal basis coefficients $w_{i,1:n}^*$ of Equation (4) are obtained, we compute the Jacobian matrix of Equation (5) at $w_{i,1:n}^*$ using the two-

point finite difference approximation in the SciPy library

$$J_i = \frac{\partial r(w_{i,1:n})}{\partial w_{i,1:n}} \Big|_{w_{i,1:n}=w_{i,1:n}^*}. \quad (6)$$

We then estimate the parameter covariance matrix using the linear approximation (K. W. Vugrin et al. 2007)

$$\Sigma_i = \begin{cases} \frac{r(w_{i,1:n}^*) r(w_{i,1:n}^*)^T}{d-n} (J_i^T J_i)^{-1}, & \text{(a)} \\ \frac{r(w_{i,1:n}^*) r(w_{i,1:n}^*)^T}{d} (J_i^T J_i)^{-1}, & \text{(b)} \end{cases} \quad (7)$$

and according to Equations (1) and (2), we transform Equation (7) into the covariance matrix of the original stellar parameters

$$\Sigma'_i = \begin{cases} Q z_{1:n} \Sigma_i z_{1:n}^T Q^T, & \text{(a)} \\ Q z \Sigma_i z^T Q^T, & \text{(b)} \end{cases}$$

where Q is the diagonal matrix formed by $\max(\theta_{\text{raw}}^{\text{train}}) - \min(\theta_{\text{raw}}^{\text{train}})$ in Equation (1). Finally, the stellar-parameter errors for the i th spectrum are the square roots of the diagonal elements of the covariance matrix Σ'_i .

For large survey applications, we control the number of concurrent processes in Joblib via the parameter `n_jobs`. Following Figure 1 of J.-C. Liang et al. (2026), we set `n_jobs=-1`, i.e., using all available CPU cores; this setting exhibits near-optimal inference efficiency across various hardware configurations.

3.2.2. PISP-Adam

PISP-Adam is designed for stellar-parameter inference in large spectroscopic datasets. Its overall workflow comprises the following steps:

1. *Batch spectrum storage.* To accommodate large-scale inference in a GPU environment, every 20,000 spectra from Section 2.3 are packaged into a single PyTorch tensor file (`.pt`). Each `.pt` file contains the normalized spectra, the official pixel masks, and the parameter initial values generated in the same way as for PISP-CurveFit. This design enables fast loading of the required data and initial values when jointly inferring $N \leq 20,000$ spectra with 25 stellar parameters each, facilitating subsequent optimization. During inference, to balance computational efficiency and GPU memory usage, the default N is set according to the type of spectral emulator: $N = 10,000$ for the FNN emulator and $N = 5000$ for the ResNet emulator. This setting follows the relationship between N and inference efficiency shown in Figure 1 of J.-C. Liang et al. (2026); throughput is typically higher when N approaches the upper bound of available GPU memory capacity.
2. *Batch reconstruction of standardized stellar parameters.* From Equation (2), we obtain the batch reconstruction of

¹³ The nearest-neighbor search is implemented using `sklearn.neighbors.KNeighborsRegressor(n_neighbors=1)`.

standardized stellar parameters for any N spectra

$$\theta_{\text{Adam}} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_N \end{pmatrix} = \begin{pmatrix} w_{1,1:n} \cdot z_{1:n}^T + \mu_\theta \\ \vdots \\ w_{N,1:n} \cdot z_{1:n}^T + \mu_\theta \end{pmatrix}, \quad (8)$$

where θ_{Adam} is the reconstructed $N \times 25$ matrix of standardized stellar parameters. The reconstruction is implemented in PyTorch with batched computation to improve efficiency.

3. *Define the objective function.* For any N spectra $F = (F_1, \dots, F_N)$ ($N \times m_1$, where $m_1 = 7214$ is the total flux dimensionality) and their pixel-mask matrix $A = (A_1, \dots, A_N)$ ($N \times m_1$), we define the objective function in the standardized stellar-parameter space

$$\mathcal{L}_{\text{Adam}}^{\text{raw}} = \sqrt{\frac{1}{N \cdot m_1} \sum_{i,j} (F_{i,j} \cdot A_{i,j} - g(\theta_{\text{Adam}})_{i,j} \cdot A_{i,j})^2}, \quad (9)$$

where

$$g(\theta_{\text{Adam}}) = \begin{pmatrix} g(\theta_1) \\ \vdots \\ g(\theta_N) \end{pmatrix},$$

and i, j index the j th wavelength pixel of the i th spectrum. Substituting Equation (8) into Equation (9) and adding L1 regularization yields the objective function in terms of basis coefficients:

$$\mathcal{L}_{\text{Adam}}^{\text{PISP}} = \sqrt{\frac{1}{N \cdot m_1} \sum_{i,j} (e_{i,j})^2 + \frac{\alpha}{N \cdot m_1} \cdot \sum_{i,t} |w_{i,t}|}, \quad (10)$$

where

$$e_{i,j} = F_{i,j} \cdot A_{i,j} - g(w_{i,1:n} \cdot z_{1:n}^T + \mu_\theta)_j \cdot A_{i,j},$$

and $\alpha \geq 0$ is the regularization coefficient. As in PISP-CurveFit, when $\alpha = 0$, the method corresponds to the PCA-Non-L1 and AS-Non-L1 strategies, in which only the first n columns of basis coefficients $w_{i,1:n}$ ($i = 1, \dots, N$) for all N samples are optimized; when $\alpha > 0$, the method corresponds to the PCA-L1 and AS-L1 strategies, for which we set $n = 25$ and optimize all basis coefficients $w_{i,1:25}$ ($i = 1, \dots, N$). Here, n is the hyperparameter for the Non-L1 strategies, and α is the hyperparameter for the L1 strategies; their determination is described in Section 4.2.1. To improve inference efficiency, both model spectrum generation and objective function computation are implemented using PyTorch for batch parallel processing.

4. *Minimization of the objective function.* PISP-Adam supports four inference strategies: PCA-Non-L1 ($\alpha = 0$), PCA-L1 ($\alpha > 0$), AS-Non-L1 ($\alpha = 0$), and AS-L1 ($\alpha > 0$). All four strategies iteratively solve for the minimizer of the objective function in Equation (10) using the Adam optimizer (D. P. Kingma & J. Ba 2014), implemented via `torch.optim.Adam`. The default learning rate is set to 0.01, and the initial values are the same as in PISP-CurveFit. We define three convergence criteria: reaching 10,000 iterations, a change in the objective function smaller than 10^{-8} over 50 consecutive iterations, or a monotonically increasing objective over 50 consecutive iterations. The learning rate and

convergence threshold are hyperparameters of PISP-Adam and are chosen to balance convergence stability and computational efficiency (see Section 4.2.2). Satisfaction of any criterion is taken as convergence, after which the optimal basis coefficients $w_{i,1:n}^*$ ($i = 1, \dots, N$) are substituted into Equation (8) to obtain θ_{Adam}^* , which are then transformed back to physical units via the inverse transformation of Equation (1). The computation of parameter errors follows the same procedure as in PISP-CurveFit.

3.3. Baseline Strategy: Direct Optimization in the Standardized Stellar-parameter Space

To evaluate the performance of PISP, we design a baseline strategy: direct optimization in the standardized stellar-parameter space without orthogonal transformation. This strategy serves as the control group for both PISP-CurveFit and PISP-Adam implementations. The optimization variable is uniformly set to the standardized stellar-parameter vector θ_i (rather than basis coefficients), the objective functions are Equations (3) and (9), respectively, and the optimization algorithms are CurveFit and Adam, respectively.

To ensure a fair comparison, this strategy adopts the same initialization as PISP: the initial values are uniformly set to the standardized stellar parameters corresponding to the nearest-neighbor spectrum of the target spectrum in the training set. The convergence criteria during optimization are also consistent with PISP. The final standardized parameters θ_i^* are transformed back to stellar parameters with physical units via the inverse transformation of Equation (1). Parameter errors are computed using the same method as PISP: the Jacobian matrix is computed at the optimum θ_i^* using two-point finite differences, and the covariance matrix of the standardized stellar parameters is estimated using linear approximation, which is then transformed to the original parameter space according to Equation (1); the square roots of the diagonal elements are taken as the final stellar-parameter errors.

This direct optimization strategy is not part of the PISP framework itself but is rather a baseline strategy without orthogonal transformation or sparsity constraints, which provides a reference for evaluating PISP's convergence speed and inference accuracy in high-dimensional space. In the following sections, we present experimental comparisons to quantitatively demonstrate the differences between this strategy and PISP in inference accuracy and efficiency.

4. Experiment

This section systematically evaluates the performance of PISP on both theoretical and observed spectra. First, we quantify the flux-reconstruction errors of the two types of spectral emulators (FNN and ResNet) on the test set to verify their fitting capabilities. Second, we separately tune PISP's key hyperparameters on the theoretical validation set and the observational reference set and select the optimal projection basis construction method. After determining the optimal settings, we assess PISP's parameter inference accuracy relative to the baseline strategy (direct optimization in the standardized stellar-parameter space) on both theoretical spectra (test set) and APOGEE DR17 observed spectra. We examine the validity of the error model by comparing the

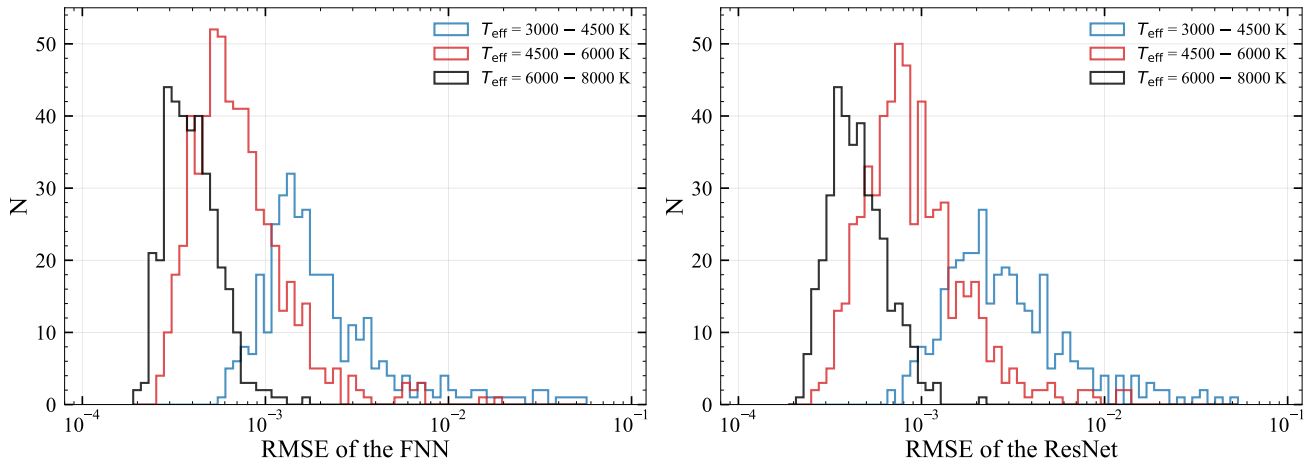


Figure 1. Distribution of spectral flux-reconstruction errors (RMSE) across different T_{eff} ranges. The left and right panels show the RMSE histograms of the FNN- and ResNet-based spectral emulators on the test set, respectively.

consistency between model errors and repeat-observation errors and also evaluate whether PISP’s inference accuracy depends on the choice of optimizer by testing multiple optimization algorithms. Finally, we assess PISP’s computational efficiency and discuss potential directions for improvement.

4.1. Evaluating the Performance of the Spectral Emulator

To assess the fitting capability of the spectral emulators, we compute, on the test set, the flux-reconstruction error for each normalized spectrum, i.e., the root mean square error (RMSE) of the flux, and we divide the sample by T_{eff} into three bins: 3000–4500, 4500–6000, and 6000–8000 K (see Figure 1). Overall, RMSE decreases with increasing T_{eff} . In the low-temperature bin (3000–4500 K), the median RMSE of the FNN is 0.0016, lower than the ResNet’s 0.0025; the fractions with $\text{RMSE} < 10^{-3}$ are 13.4% and 4.8%, respectively, indicating that the FNN is more stable for cool-star spectra. In the intermediate bin (4500–6000 K), both models perform well, with the FNN retaining a slight advantage (fraction with $\text{RMSE} < 10^{-3}$: 77.4% versus 63.1%). In the high-temperature bin (6000–8000 K), the median RMSE of both models drops to about 0.0004, and the fraction with $\text{RMSE} < 10^{-3}$ exceeds 98% for both, indicating essentially comparable reconstruction errors. Given that the two emulators use the same training set, validation strategy, and ReduceLROnPlateau scheduler, and that the final models are selected at the minimum validation loss, the observed differences are unlikely to originate from the training procedure and more likely reflect architectural effects. It is worth noting that the absolute difference between the median RMSEs is less than 10^{-3} , and no significant difference is observed in subsequent parameter inference. We employ two structurally different emulators—FNN and ResNet—to span different inductive biases and representational capacities, ensuring that the conclusions about PISP do not depend on a single architecture, rather than to compare the emulators themselves.

4.2. Configuration of PISP

PISP requires a predefined set of projection bases for the inference stage and performs optimization within the corresponding projection space. Consequently, the parameter

inference results depend on both the basis construction method and the associated hyperparameter settings. This work implements and compares two basis construction methods: PCA-based and AS-based. Beyond the choice of basis construction, PISP’s configuration parameters can be functionally categorized into two groups: (1) projection-space-related parameters, including the number of retained projection dimensions n in the PCA-Non-L1 and AS-Non-L1 strategies, and the regularization coefficient α in the PCA-L1 and AS-L1 strategies, which suppress the influence of certain projection directions on parameter inference and require independent tuning for different datasets; and (2) optimizer-related parameters, including the learning rate lr and convergence threshold ϵ , which primarily affect the stability and convergence speed of the iterative process and generally exhibit strong transferability across datasets. Based on this division, we adopt a two-stage tuning strategy. In the first stage, using the Kurucz theoretical spectrum validation set and the APOGEE DR17 reference subset (see Section 4.2.1), we employ the FNN spectral emulator with PISP-CurveFit to separately tune n and α for both PCA and AS bases, thereby determining the default basis construction method and its optimal (n, α) configuration for subsequent experiments. In the second stage, with the default basis configuration and optimal (n, α) established, we further tune the lr and ϵ of PISP-Adam on the theoretical spectrum validation set to obtain more robust optimizer settings across various inference tasks.

4.2.1. Determination of Optimal Projection Dimension n , Regularization Coefficient α , and Projection Basis

To determine PISP’s key configuration—the basis construction method (PCA or AS) and hyperparameters (n, α) —we first define the evaluation metric: for each setting, we compute the standard deviation of the differences ($\sigma(\Delta)$) between the inferred 25-dimensional labels and the reference labels, and select as optimal the setting that minimizes the sum of standard deviations across all dimensions ($\sum \sigma(\Delta)$). Subsequently, under both Non-L1 and L1 settings, we compare the PCA and AS basis constructions and select the superior method as the default. To accommodate different datasets, we tune on both the Kurucz validation set and an APOGEE reference subset:

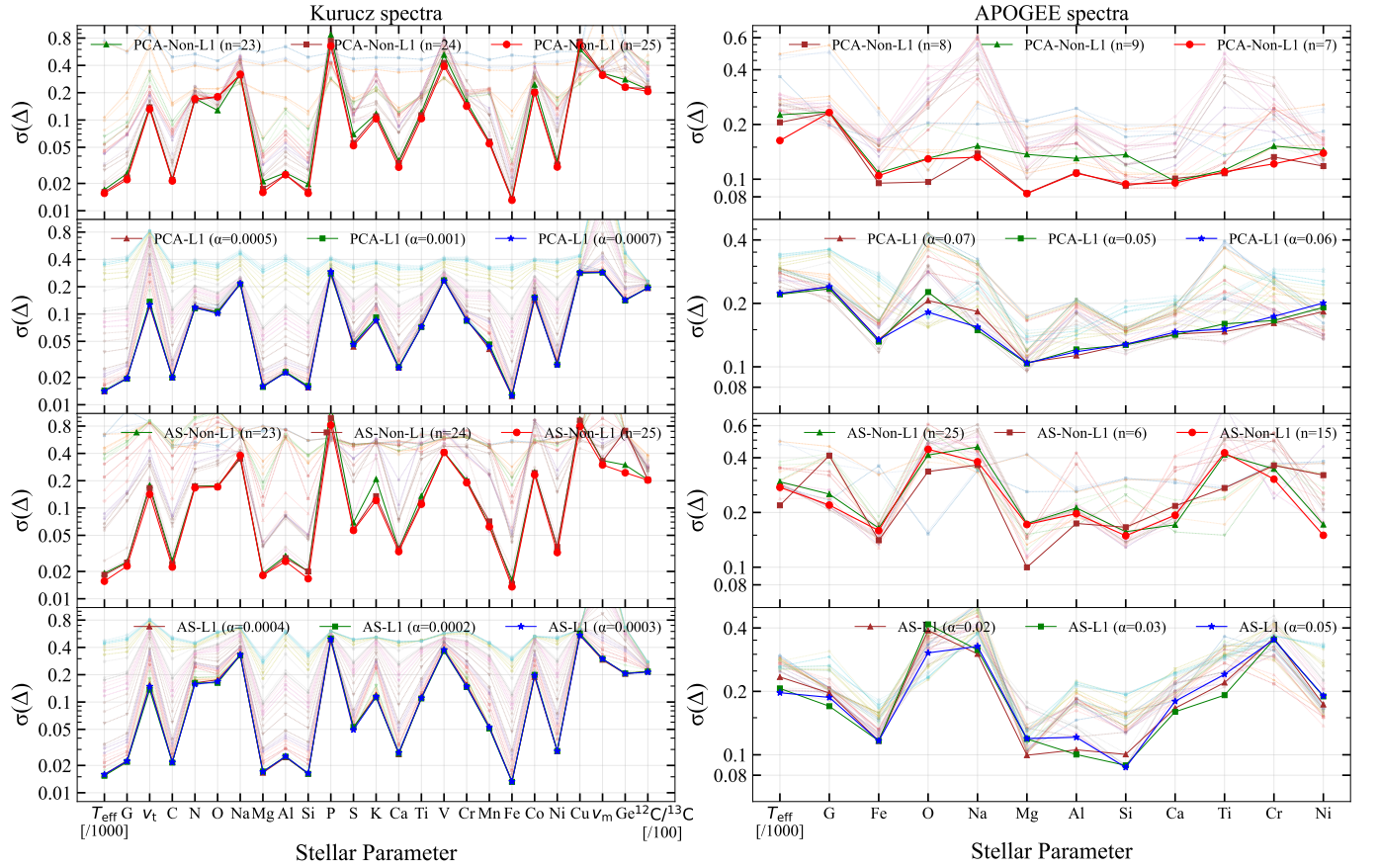


Figure 2. Hyperparameter settings (n and α) of PISP across different datasets. The first column lists the hyperparameters tuned on the Kurucz validation set, while the second column shows those optimized using the APOGEE DR17 reference set. The x -axis indicates the stellar parameters and the y -axis shows the standard deviation (σ) of the differences (Δ) between inferred and reference values, with σ computed using `astropy.stats.sigma_clipped_stats(sigma=3, maxiters=3)`. From top to bottom, the four rows correspond to the PCA-Non-L1, PCA-L1, AS-Non-L1, and AS-L1 strategies, respectively. In the PCA-Non-L1 and AS-Non-L1 panels, the red curve denotes the optimal projection dimension n , while in the PCA-L1 and AS-L1 panels, the blue curve denotes the optimal α . The lighter curves in each panel show the results for all other tested hyperparameter values (e.g., in the first-row first-column panel, all cases with $n < 23$ are shown). The y -axis adopts a base-10 logarithmic scale, with minor ticks indicating evenly spaced subdivisions between adjacent major ticks and tick labels showing the corresponding linear values. For clarity, $\log g$, V_{turb} , and V_{macro} are abbreviated as G , V_t , and V_m along the x -axis.

1. Hyperparameter tuning based on the Kurucz validation set.

a. *Determining n for PCA-Non-L1 and AS-Non-L1 strategies.* Varying n from 1 to 25, we run inference for each setting. The subplots in row 1, column 1 (PCA-Non-L1) and row 3, column 1 (AS-Non-L1) of Figure 2 show the standard deviations of parameter differences $\sigma(\Delta)$ for different values of n . When $n < 23$, the $\sigma(\Delta)$ is generally larger, indicating that basis vectors corresponding to smaller eigenvalues in either PCA or AS bases may still contain information useful for spectral fitting and thus cannot be neglected. Further comparison using the $\sum \sigma(\Delta)$ metric shows that this metric reaches its minimum at $n = 25$. Therefore, we select $n = 25$ as the optimal projection dimension for both PCA-Non-L1 and AS-Non-L1 strategies on Kurucz theoretical spectra.

b. *Determining α for PCA-L1 and AS-L1 strategies.* With $n = 25$ fixed, we conduct a grid search over $\alpha \in \{1\} \cup \{a \times 10^{-k} | a = 1, \dots, 9; k = 1, \dots, 4\}$. As shown in row 2, column 1 (PCA-L1) and row 4, column 1 (AS-L1) of Figure 2, different α values significantly impact $\sigma(\Delta)$, with smaller α generally yielding better performance: for instance, under the

PCA-L1 strategy, $\alpha = 10^{-3}$, 7×10^{-4} , and 5×10^{-4} produce superior results; while under the AS-L1 strategy, $\alpha = 4 \times 10^{-4}$, 2×10^{-4} , and 3×10^{-4} exhibit better performance. This indicates that excessive regularization imposes overly stringent constraints on the solution space, thereby limiting the model's ability to fit spectral details. Further comparison using the $\sum \sigma(\Delta)$ metric shows that PCA-L1 achieves optimality at $\alpha = 7 \times 10^{-4}$, while AS-L1 is optimal at $\alpha = 3 \times 10^{-4}$.

2. *Hyperparameter tuning based on the APOGEE reference subset.* To address distributional differences between synthetic and observed spectra, we retune n and α on APOGEE spectra. To avoid potential label leakage from the APOGEE catalog, we use the independent high-resolution analysis of T. Bensby et al. (2014) as a reference, which provides precise parameters for 714 F/G dwarfs from spectra with $R = 40,000\text{--}110,000$ and signal-to-noise ratio (S/N) > 150 . Cross-matching with APOGEE DR17 yields 45 stars, covering 12 of the 25 labels considered here. We use the same hyperparameter grid as for the synthetic data. As shown in column 2 of Figure 2, the optimal projection dimension for PCA-Non-L1 falls within $n \in \{7, 8, 9\}$, and the optimal

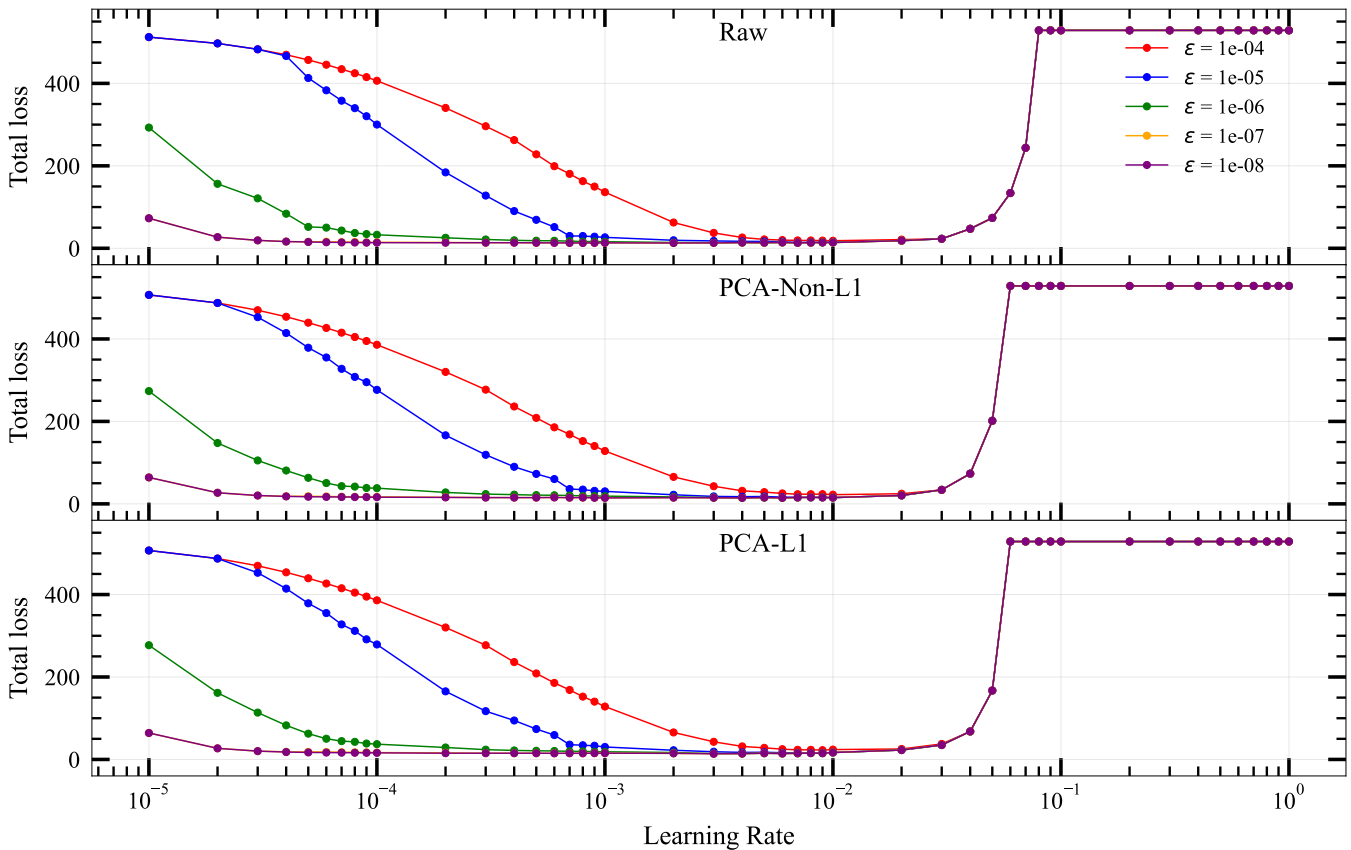


Figure 3. Hyperparameter settings (lr and ϵ) of PISP-Adam on the Kurucz validation set. The three panels correspond to optimization in the baseline strategy (Raw), PCA-Non-L1 ($n = 25$), and PCA-L1 ($\alpha = 7 \times 10^{-4}$), respectively. Each curve represents a fixed convergence threshold ϵ ranging from 10^{-8} to 10^{-4} . The x-axis adopts a base-10 logarithmic scale to represent the learning rate lr , with tick labels indicating the corresponding linear values and minor ticks evenly spaced between adjacent major ticks.

regularization coefficient for PCA-L1 falls within $\alpha \in \{0.05, 0.06, 0.07\}$; correspondingly, the optimal n for AS-Non-L1 falls within $\{6, 15, 25\}$, and the optimal α for AS-L1 falls within $\{0.02, 0.03, 0.05\}$. These results differ from the optimal settings for Kurucz spectra, reflecting distributional differences between theoretical and observed data; such biases can be effectively mitigated by reducing the projection dimension or increasing regularization strength. Further comparison using the $\sum\sigma(\Delta)$ metric leads to the final selection of $n = 7$ (PCA-Non-L1) and $\alpha = 0.06$ (PCA-L1), as well as $n = 15$ (AS-Non-L1) and $\alpha = 0.05$ (AS-L1) as the optimal hyperparameter settings for APOGEE.

3. *Determining the optimal projection basis construction method.* After determining the optimal projection dimensions n and regularization coefficients α for both PCA and AS bases on theoretical and observed spectra, we use the $\sum\sigma(\Delta)$ metric to select the default projection basis construction method for subsequent experiments. As shown in Table 2, PCA-Non-L1 outperforms AS-Non-L1 and PCA-L1 outperforms AS-L1 on both Kurucz theoretical spectra and the APOGEE reference subset. Therefore, in subsequent experiments (under both Non-L1 and L1 settings), we adopt PCA projection bases as the default.

Table 2
Comparison of PCA and AS Projection Bases for PISP with Optimal Hyperparameter Configurations

Strategy	Dataset	n	α	$\sum\sigma(\Delta)$
PCA-Non-L1	Kurucz	25	0	4.12
AS-Non-L1	Kurucz	25	0	4.60
PCA-L1	Kurucz	25	0.0007	2.65
AS-L1	Kurucz	25	0.0003	3.80
PCA-Non-L1	APOGEE	7	0	1.51
AS-Non-L1	APOGEE	15	0	3.07
PCA-L1	APOGEE	25	0.06	1.95
AS-L1	APOGEE	25	0.05	2.42

Note. The minimum $\sum\sigma(\Delta)$ between PCA and AS within each dataset and regularization setting is highlighted in boldface. The definition of $\sigma(\Delta)$ is given in Figure 2.

4.2.2. Learning Rate lr and Convergence Threshold ϵ in PISP-Adam

To determine the Adam optimizer hyperparameters lr and ϵ in PISP-Adam, we fix $n = 25$ (PCA-Non-L1) and $\alpha = 7 \times 10^{-4}$ (PCA-L1) and evaluate the aggregate validation loss on the Kurucz validation set across different (lr , ϵ) combinations.

Table 3
Standard Deviations of Stellar Parameter Differences with Respect to Kurucz Reference Labels on the Test Set

Parameter	FNN						ResNet					
	CurveFit			Adam			CurveFit			Adam		
	Raw	Non-L1	L1	Raw	Non-L1	L1	Raw	Non-L1	L1	Raw	Non-L1	L1
T_{eff}	16	16	14	16	16	15	15	15	14	15	15	14
$\log g$	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
v_{turb}	0.14	0.13	0.13	0.14	0.13	0.13	0.12	0.13	0.13	0.13	0.12	0.13
[C/H]	0.02	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
[N/H]	0.19	0.17	0.11	0.20	0.17	0.11	0.15	0.14	0.10	0.15	0.13	0.10
[O/H]	0.22	0.16	0.09	0.20	0.14	0.10	0.14	0.12	0.08	0.13	0.12	0.08
[Na/H]	0.37	0.30	0.21	0.34	0.29	0.22	0.28	0.22	0.15	0.27	0.22	0.16
[Mg/H]	0.02	0.02	0.01	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
[Al/H]	0.03	0.03	0.02	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02
[Si/H]	0.02	0.02	0.01	0.02	0.02	0.01	0.02	0.02	0.01	0.02	0.01	0.01
[P/H]	1.02	0.72	0.30	0.72	0.42	0.31	0.88	0.68	0.32	0.82	0.47	0.31
[S/H]	0.06	0.05	0.05	0.07	0.06	0.05	0.05	0.05	0.04	0.05	0.05	0.04
[K/H]	0.11	0.10	0.09	0.15	0.11	0.09	0.07	0.06	0.07	0.07	0.07	0.07
[Ca/H]	0.03	0.03	0.03	0.04	0.04	0.03	0.03	0.03	0.02	0.03	0.03	0.02
[Ti/H]	0.11	0.10	0.07	0.14	0.10	0.08	0.09	0.09	0.07	0.10	0.09	0.07
[V/H]	0.49	0.34	0.22	0.40	0.33	0.24	0.56	0.44	0.24	0.52	0.35	0.25
[Cr/H]	0.17	0.14	0.08	0.14	0.12	0.09	0.09	0.10	0.07	0.10	0.09	0.07
[Mn/H]	0.05	0.05	0.04	0.06	0.05	0.04	0.04	0.04	0.04	0.05	0.05	0.04
[Fe/H]	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
[Co/H]	0.22	0.19	0.14	0.23	0.19	0.14	0.30	0.28	0.17	0.31	0.24	0.17
[Ni/H]	0.03	0.03	0.03	0.03	0.03	0.03	0.05	0.04	0.03	0.05	0.04	0.03
[Cu/H]	0.97	0.77	0.29	0.75	0.36	0.29	0.39	0.32	0.23	0.39	0.29	0.23
v_{macro}	0.33	0.35	0.34	0.36	0.35	0.34	0.34	0.34	0.31	0.36	0.34	0.31
[Ge/H]	0.21	0.17	0.14	0.20	0.19	0.14	0.15	0.15	0.13	0.16	0.14	0.13
$^{12}\text{C}/^{13}\text{C}$	23	21	19	20	19	18	27	22	18	22	19	18

Note. Values are σ computed using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`) for differences ($\Delta = \text{inferred} - \text{reference}$). For brevity, “Raw” denotes the baseline strategy, “Non-L1” refers to PCA-Non-L1 ($n = 25$), and “L1” refers to PCA-L1 ($\alpha = 0.0007$). Within each emulator–optimizer pair, the minimum σ among the three strategies is highlighted in boldface.

We test three strategies: (1) the baseline of direct optimization in the standardized parameter space, (2) PCA-Non-L1, and (3) PCA-L1. For each strategy, we set $\epsilon \in \{10^{-i} | i = 4, 5, 6, 7, 8\}$; for each ϵ we scan $lr \in \{1\} \cup \{a \times 10^{-k} | a = 1, \dots, 9; k = 1, \dots, 5\}$. Figure 3 shows the distribution of aggregate validation loss over (lr, ϵ) . The results indicate that, for a fixed lr , smaller ϵ generally yields lower loss, with the minimum at $\epsilon = 10^{-8}$; all three strategies exhibit good convergence for $lr \in [0.001, 0.01]$, suggesting this is a robust range. Furthermore, at $\epsilon = 10^{-8}$, the runtime with $lr = 0.001$ is approximately $6\times$, $7\times$, and $6\times$ that with $lr = 0.01$ for the baseline, PCA-Non-L1, and PCA-L1, respectively, indicating that $lr = 0.01$ substantially reduces inference cost while maintaining low loss. Balancing convergence and efficiency, we recommend $lr = 0.01$ and $\epsilon = 10^{-8}$ as default settings for PISP-Adam to support stable and efficient large-scale inference.

4.3. Parameter Inference on Synthetic Spectra

To evaluate the performance of PISP-CurveFit and PISP-Adam on the theoretical spectrum test set, we adopt the optimal PISP configuration determined in Section 4.2—PCA projection bases with hyperparameters ($n = 25$, $\alpha = 7 \times 10^{-4}$, $lr = 0.01$, $\epsilon = 10^{-8}$)—and compare the two PCA-assisted strategies (PCA-Non-L1 and PCA-L1) with the baseline strategy.

Table 3 presents the standard deviation of parameter differences $\sigma(\Delta)$ for two spectral emulators and two optimizers under three inference strategies (Raw, PCA-Non-L1, PCA-L1). Across most label dimensions, PCA-L1 performs best: 12 of 20 elemental abundances show $\sigma(\Delta)$ reductions of at least 0.01 dex relative to the baseline strategy in all four emulator–optimizer combinations, with [N/H], [O/H], [Na/H], [Co/H], [P/H], [V/H], and [Cu/H] showing reductions of 0.05–0.72 dex. Consider the case of FNN+Adam: PCA-L1 yields notably lower $\sigma(\Delta)$ than the baseline strategy for multiple elemental abundances—[P/H] decreases from 0.72 to 0.31 dex, [Cu/H] from 0.75 to 0.29 dex, [V/H] from 0.40 to 0.24 dex, while [N/H] and [O/H] decrease from 0.20 dex to 0.11 dex and 0.10 dex, respectively. For parameters with low baseline $\sigma(\Delta)$ (e.g., $\log g$ and [Fe/H]), differences among strategies are minimal. Figure 4 illustrates this comparison for FNN+Adam, showing inferred versus reference labels under both the baseline and PCA-L1 strategies. Relative to the baseline strategy, PCA-L1 reduces $\sigma(\Delta)$ by 25%–61% for abundance labels, and systematic biases show minimal variation. Similar trends emerge across other emulator–optimizer combinations, demonstrating consistent improvements.

We further evaluate PISP performance across six $T_{\text{eff}} - \log g$ subregions (Figures 5 and 6). For both FNN and ResNet emulators, PISP reduces $\sigma(\Delta)$ for most elemental abundances relative to the baseline strategy. Consider PCA-L1 with FNN+Adam (Figure 5): defining $\Delta\sigma = \sigma_{\text{Raw}} - \sigma_{\text{PCA-L1}}$, we adopt

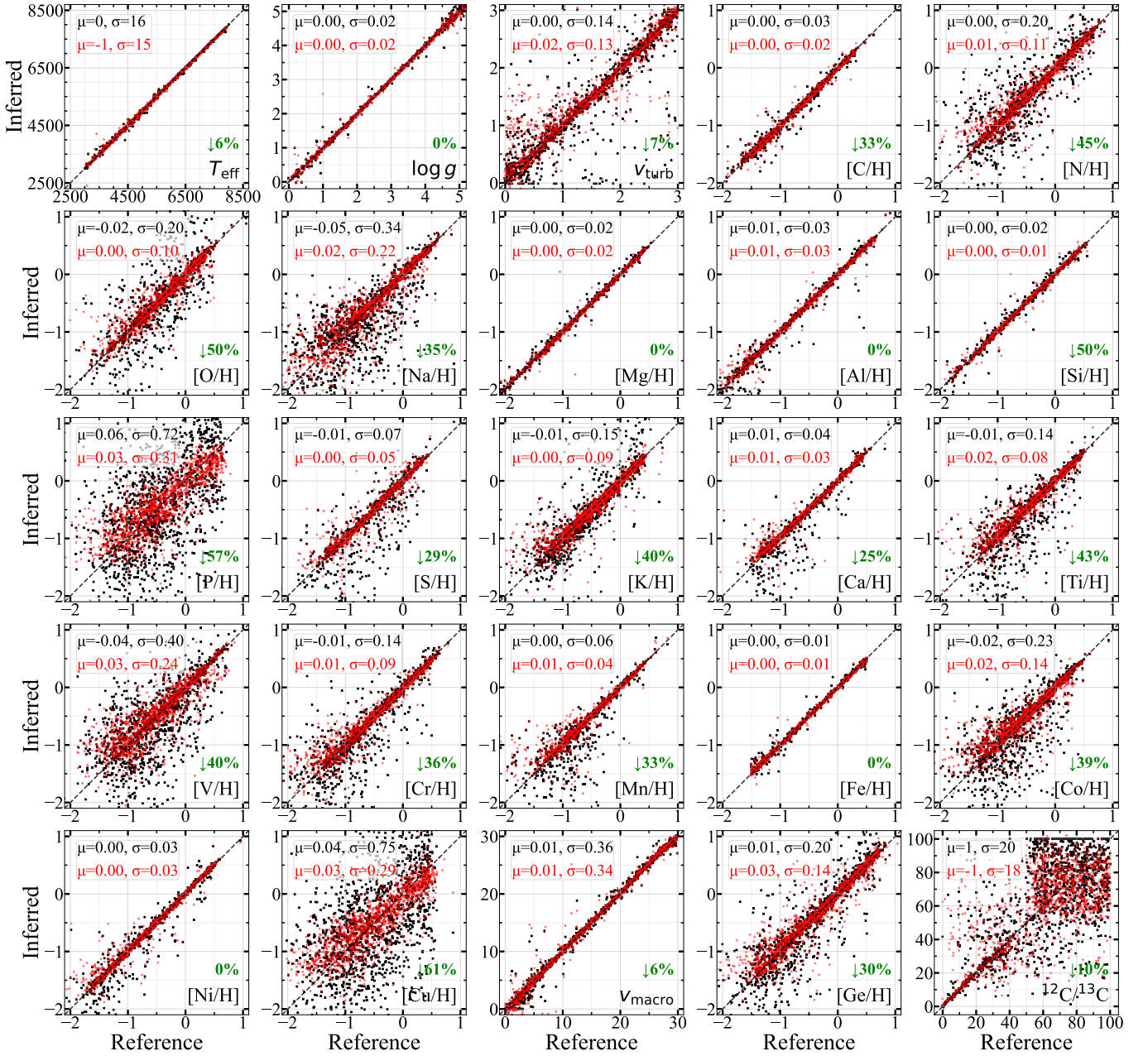


Figure 4. Comparison of stellar parameters inferred by PISP-Adam and the baseline against Kurucz reference labels on the test set. All results are based on the FNN spectral emulator and the Adam optimizer. Each panel shows inferred versus reference labels: black squares correspond to the baseline strategy, while red circles represent the PCA-L1 strategy ($\alpha = 7 \times 10^{-4}$). Differences ($\Delta = \text{inferred} - \text{reference}$) statistics (μ, σ) were computed via `astropy.stats.sigma_clipped_stats` (`sigma=3, maxiters=3`). Green numbers indicate the relative σ reduction achieved by PCA-L1. The statistics are listed in the upper left in matching colors.

$\Delta\sigma \geq 0.01$ dex as the criterion for “effective improvement” in elemental abundances. In the cool-temperature regime ($3000 < T_{\text{eff}} < 4500$), 7 ($\log g < 3.5$) and 15 ($\log g \geq 3.5$) elemental abundances satisfy this criterion; in the intermediate-temperature regime ($4500 \leq T_{\text{eff}} < 6000$), the corresponding numbers are 9 ($\log g < 3.5$) and 11 ($\log g \geq 3.5$); in the hot-temperature regime ($T_{\text{eff}} \geq 6000$), the numbers increase to 14 ($\log g < 3.5$) and 15 ($\log g \geq 3.5$). Elements showing “effective improvement” across all six subregions include [Na/H], [P/H], [Cu/H], [V/H], and [Ge/H], with $\Delta\sigma$ ranges of 0.05–0.19 dex, 0.19–0.53 dex, 0.33–0.57 dex, 0.02–0.26 dex, and 0.02–0.16 dex, respectively. Additionally, “effective improvement” for certain elements exhibit

temperature dependence: for instance, improvements for [N/H] and [O/H] primarily occur in the hot-temperature regime, with $\Delta\sigma_{[\text{N}/\text{H}]} = 0.28\text{--}0.32$ dex and $\Delta\sigma_{[\text{O}/\text{H}]} = 0.36\text{--}0.40$ dex; while [Ti/H] and [Cr/H] also achieve $\Delta\sigma_{[\text{Ti}/\text{H}]} = 0.21\text{--}0.28$ dex and $\Delta\sigma_{[\text{Cr}/\text{H}]} = 0.21\text{--}0.29$ dex in the hot regime. In contrast, for parameters such as T_{eff} and $\log g$, the improvements in σ are on a smaller scale (e.g., $\Delta\sigma_{T_{\text{eff}}} \leq 5$ K, $\Delta\sigma_{\log g} \leq 0.002$ dex). In a few cases, the baseline strategy performs better, such as in the subregion with $3000 < T_{\text{eff}} < 4500$ and $\log g < 3.5$, where [O/H], [Al/H], [Ti/H], and [Cr/H] correspond to $\Delta\sigma$ values of -0.003 , -0.003 , -0.001 , and -0.005 dex, respectively. These results demonstrate that PISP primarily enhances precision for

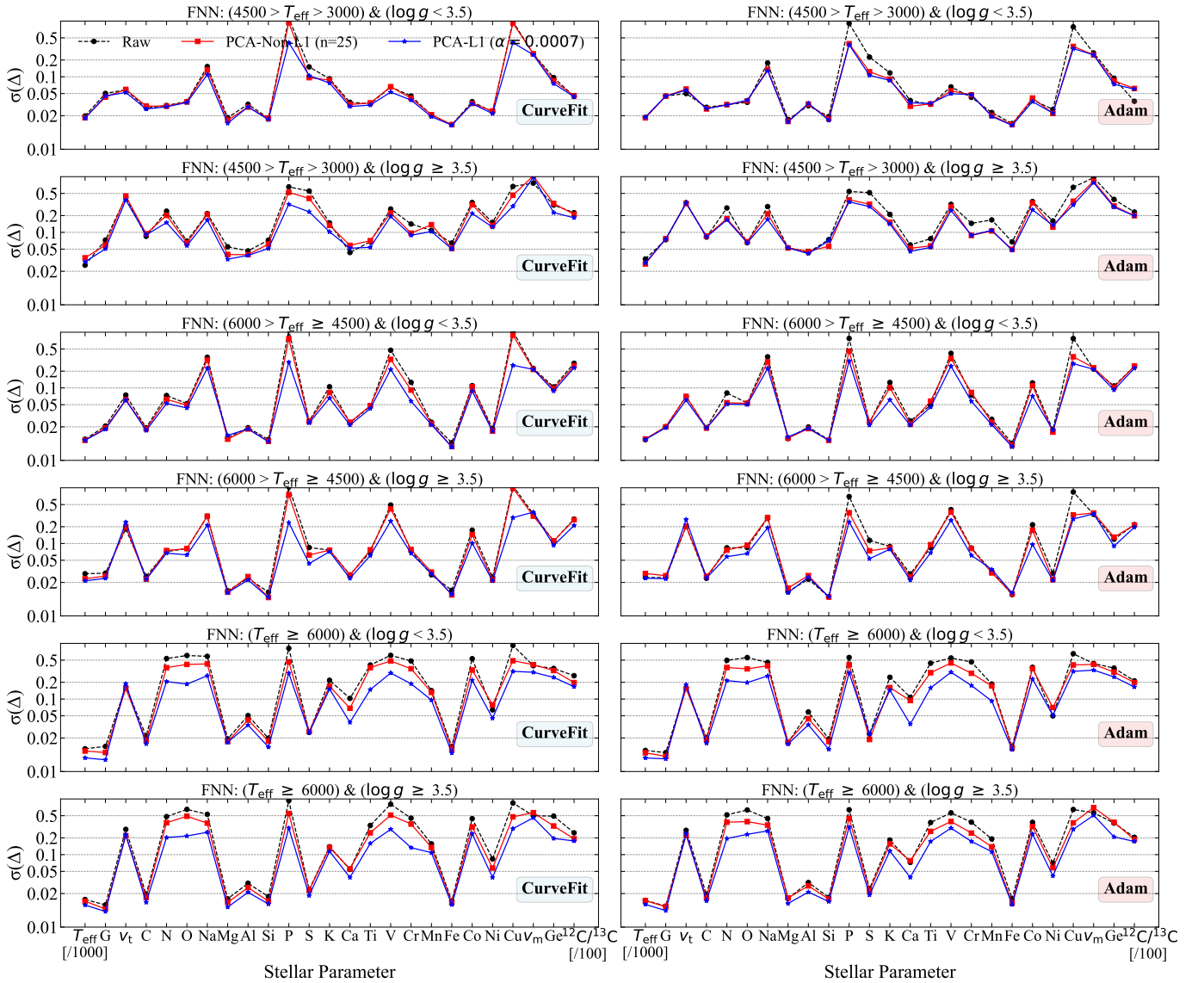


Figure 5. Standard deviations of differences relative to the Kurucz reference labels for PISP and baseline inferences on the Kurucz test set using the FNN spectral emulator. The 12 panels are organized into six $T_{\text{eff}}\text{-log } g$ subsets (rows), and the optimizer changes by column (left: CurveFit; right: Adam). In each panel, σ is computed from differences ($\Delta = \text{inferred} - \text{reference}$) for 25 labels using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`). For clarity, the legend is shown only in the first panel: the black, red, and blue curves correspond to the baseline strategy described in Section 3.3, the PCA-Non-L1 strategy with $n = 25$, and the PCA-L1 strategy with $\alpha = 0.0007$, respectively, each is evaluated at its optimal hyperparameter setting. On the x-axis, $\log g$, V_{urb} , and V_{macro} are denoted G , V_t , and V_m ; T_{eff} is shown in units of 10^3 K, and $^{12}\text{C}/^{13}\text{C}$ is scaled by 100. The y-axis is base-10 logarithmic with tick labels given in linear units, and the dashed horizontal lines indicate reference levels to aid visual comparison across labels.

elemental abundances, with consistent improvements across temperature and gravity regimes, though isolated cases exhibit $\Delta\sigma \leq 0$.

Notably, although PCA-Non-L1 improves inference accuracy over the baseline strategy, it still underperforms PCA-L1 for most labels (see Table 3, or the red and blue solid lines in Figures 5 and 6). Comparing weight distributions across basis vectors (Figure 7) reveals the cause: PCA-Non-L1 assigns lower weights to low-variance basis vectors (higher fraction of $|w| < 10^{-3}$), whereas PCA-L1 concentrates weights on the first and last seven basis vectors while suppressing intermediate ones. This indicates that when training and test distributions are consistent (both from Kurucz synthetic spectra), low-variance basis vectors carry information crucial for parameter inference, and PCA-L1’s ability to adaptively select relevant basis vectors yields superior performance compared to PCA-

Non-L1. We therefore recommend PCA-L1 as the default for matched distributions.

4.4. Parameter Inference on Observed Spectra

To assess PISP’s applicability to real observational data, we apply PISP-CurveFit and PISP-Adam to APOGEE DR17 observed spectra, adopting the optimal PISP configuration determined in Section 4.2—PCA projection bases with hyperparameters: $n = 7$, $\alpha = 0.06$, $lr = 0.01$, $\epsilon = 10^{-8}$. We compare against APOGEE official catalog labels, selecting samples with $T_{\text{eff}} \in [3000, 8000]$ K and `STAR_FLAG`, `RV_FLAG`, `ASPCAP_FLAG`, as well as individual parameter `FLAG` values, all equal to zero.

Table 4 presents the standard deviations of parameter differences $\sigma(\Delta)$ relative to APOGEE official catalog labels for two spectral emulators and two optimizers under three

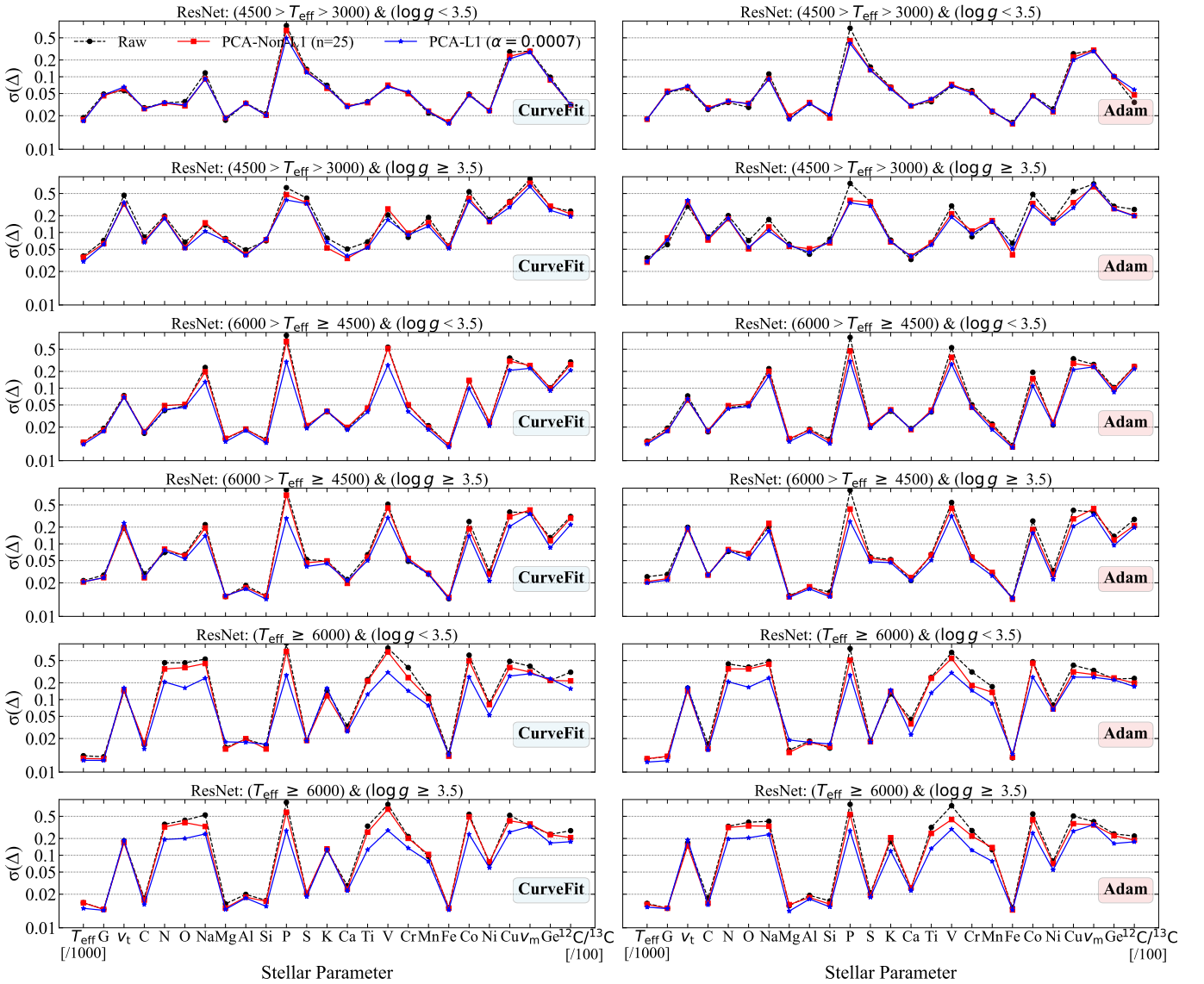


Figure 6. Standard deviations of differences relative to the Kurucz reference labels for PISP and baseline inferences on the Kurucz test set using the ResNet spectral emulator. Same as Figure 5, but with the FNN emulator replaced by ResNet.

inference strategies (Raw, PCA-Non-L1, PCA-L1). Across most stellar-parameter dimensions, PCA-Non-L1 and PCA-L1 achieve the smallest $\sigma(\Delta)$, outperforming the baseline strategy: PCA-Non-L1 reduces $\sigma(\Delta)$ by more than 30 K for T_{eff} and by at least 0.01 dex for 9 of 17 elemental abundances in all four emulator–optimizer combinations, with [O/H], [Na/H], [V/H] showing reductions of 0.05–0.20 dex. For example, with ResNet+CurveFit, PCA-Non-L1 reduces T_{eff} from 68 to 34 K, [O/H] from 0.14 to 0.08 dex, [Na/H] from 0.34 to 0.17 dex, [Si/H] from 0.11 to 0.06 dex, [V/H] from 0.38 to 0.18 dex, and [Co/H] from 0.21 to 0.16 dex. However, the improvements are not uniform: for [Al/H], [Ca/H], and [Fe/H], the baseline strategy retains a $\sigma(\Delta)$ advantage of approximately 0.01–0.05 dex over PCA-Non-L1.

Figures 8 and 9 compare $\sigma(\Delta)$ between PISP and the baseline strategy across six $T_{\text{eff}}\text{--}\log g$ subregions. Due to FLAG restrictions in the APOGEE official catalog, the number of available reference parameters varies across subregions, ranging from 14 to 21 parameters. Overall, both PCA-L1 and PCA-Non-L1 outperform the baseline strategy for both

emulators, though their relative advantages vary across $T_{\text{eff}}\text{--}\log g$ subregions. For instance, in the giant regime at $4500 \leq T_{\text{eff}} < 6000$, PCA-L1 achieves lower $\sigma(\Delta)$ for 13 parameters while PCA-Non-L1 achieves lower $\sigma(\Delta)$ for six parameters; in the dwarf regime at $4500 \leq T_{\text{eff}} < 6000$, PCA-Non-L1 outperforms PCA-L1 for 14 parameters versus six; in the dwarf regime at $T_{\text{eff}} \geq 6000$, PCA-Non-L1 leads for 14 parameters versus four for PCA-L1, while the remaining subregions show comparable performance. Combining the global statistics from Table 4 with the $T_{\text{eff}}\text{--}\log g$ subregion analysis reveals that PCA-Non-L1 slightly outperforms PCA-L1 in the number of parameters achieving superior performance on APOGEE observed spectra, contrasting with PCA-L1’s dominance on the theoretical spectrum test set. To understand this discrepancy, we compare the weight distributions across basis vectors for both strategies. As shown in Figure 10, PCA-L1 and PCA-Non-L1 exhibit different weight allocation patterns across basis vectors. Based on this observation, we offer the following qualitative interpretation: spectral features represented by low-variance basis vectors

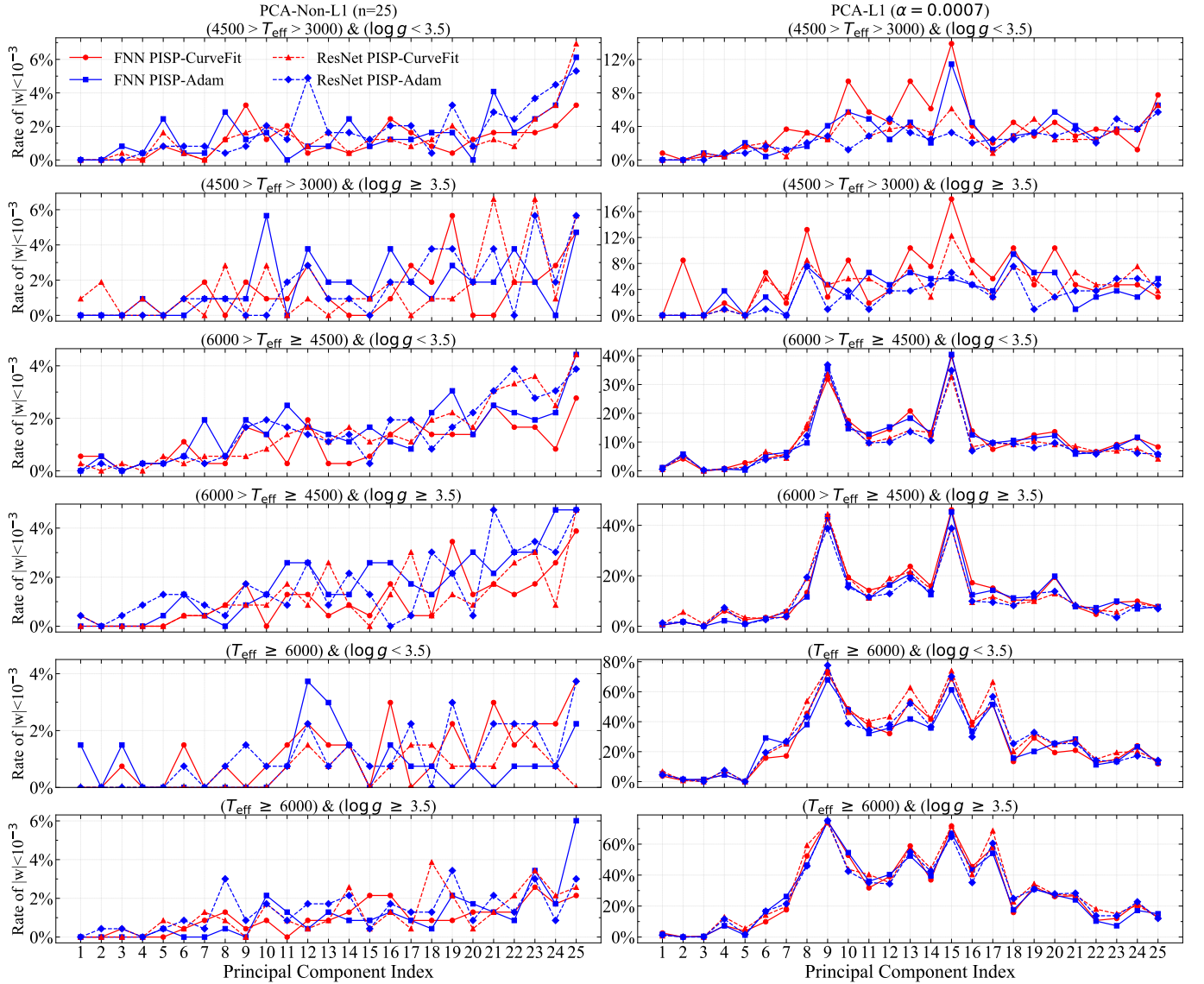


Figure 7. Fractional statistics of principal-component coefficients (w) with absolute values below 10^{-3} derived using PISP on the Kurucz test set. The first and second columns correspond to the PCA-Non-L1 and PCA-L1 strategies, respectively. The x -axis indicates the index of the principal component, and the y -axis shows the fraction of test samples with corresponding coefficients satisfying $|w| < 10^{-3}$. The larger the fraction, the less important the corresponding principal component is for parameter inference. Each panel represents a different $T_{\text{eff}}\text{-}\log g$ subrange. Solid red and blue curves denote results from FNN-based spectral emulators using CurveFit and Adam, respectively, while dashed red and blue curves represent the corresponding results from ResNet-based emulators. For clarity, the legend is shown only in the first panel.

may be more susceptible to noise and model-observation mismatches in observed data. Consequently, PCA-L1's stronger emphasis on these low-variance basis vectors may limit its precision gains, while PCA-Non-L1's focus on high-variance basis vectors may be more robust. In practice, strategy selection should be guided by data quality and the consistency between training and target distributions: when spectral noise levels are high or significant domain shifts exist, PCA-Non-L1 may be preferable; when data quality is high and distributions are well matched, PCA-L1 may still provide additional precision improvements.

4.5. Error Analysis

To evaluate the validity of PISP's model errors, we compare model errors with repeat-observation errors, using the inference results from the ResNet emulator combined with

PISP-CurveFit as an example. Model errors are derived via first-order linear error propagation in the neighborhood of the optimal solution, estimating parameter errors arising from the propagation of spectral noise from pixel space to parameter space. Repeat-observation errors originate from the dispersion of parameters across multiple independent observations of the same target, reflecting the combined effects of noise and systematic variations. Since all spectra employ consistent normalization and pixel masking strategies (Section 2.3), differences introduced by continuum processing and outlier pixels are minimized; thus, repeat-observation errors serve as an empirical reference for model errors. If model errors are reasonable, the two should be statistically consistent and exhibit similar S/N dependencies. For this analysis, we select 5939 spectra from 1049 stars out of 722,896 APOGEE spectra, where each star has at least four independent observations with S/N not exceeding 200.

Table 4
Standard Deviations of Stellar Parameter Differences with Respect to APOGEE DR17 Catalog Values

Parameter	FNN						ResNet					
	CurveFit			Adam			CurveFit			Adam		
	Raw	Non-L1	L1	Raw	Non-L1	L1	Raw	Non-L1	L1	Raw	Non-L1	L1
T_{eff}	66	36	52	75	38	59	68	34	51	73	36	56
$\log g$	0.16	0.13	0.14	0.17	0.16	0.16	0.15	0.13	0.14	0.16	0.15	0.15
v_{turb}	0.16	0.16	0.19	0.20	0.20	0.21	0.16	0.17	0.19	0.18	0.19	0.20
[C/H]	0.09	0.09	0.08	0.10	0.09	0.09	0.09	0.09	0.08	0.09	0.09	0.08
[N/H]	0.09	0.08	0.09	0.10	0.09	0.09	0.09	0.08	0.08	0.09	0.09	0.08
[O/H]	0.14	0.08	0.13	0.15	0.09	0.14	0.14	0.08	0.12	0.14	0.09	0.13
[Na/H]	0.35	0.17	0.16	0.28	0.18	0.18	0.34	0.17	0.16	0.31	0.17	0.17
[Mg/H]	0.08	0.06	0.09	0.08	0.07	0.09	0.08	0.06	0.09	0.09	0.06	0.09
[Al/H]	0.11	0.12	0.11	0.12	0.12	0.12	0.11	0.12	0.11	0.11	0.12	0.11
[Si/H]	0.11	0.06	0.12	0.11	0.07	0.12	0.11	0.06	0.12	0.11	0.07	0.12
[S/H]	0.11	0.08	0.09	0.12	0.09	0.10	0.11	0.08	0.09	0.11	0.08	0.09
[K/H]	0.15	0.13	0.11	0.16	0.13	0.12	0.13	0.13	0.11	0.14	0.13	0.11
[Ca/H]	0.06	0.07	0.06	0.07	0.07	0.07	0.05	0.07	0.06	0.06	0.07	0.06
[Ti/H]	0.13	0.11	0.12	0.14	0.11	0.13	0.13	0.11	0.12	0.14	0.11	0.12
[V/H]	0.34	0.18	0.20	0.27	0.18	0.21	0.38	0.18	0.20	0.33	0.18	0.21
[Cr/H]	0.11	0.09	0.11	0.12	0.09	0.11	0.10	0.09	0.10	0.11	0.09	0.10
[Mn/H]	0.11	0.11	0.14	0.12	0.11	0.14	0.11	0.10	0.14	0.12	0.11	0.14
[Fe/H]	0.04	0.05	0.06	0.05	0.05	0.07	0.04	0.05	0.06	0.05	0.05	0.06
[Co/H]	0.20	0.16	0.16	0.20	0.16	0.17	0.21	0.16	0.15	0.22	0.16	0.16
[Ni/H]	0.05	0.10	0.05	0.06	0.11	0.07	0.05	0.10	0.05	0.06	0.10	0.06
v_{macro}	2.18	2.29	1.64	2.40	2.43	1.98	2.11	2.10	1.52	2.21	2.17	1.59

Note. Values are σ computed using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`) for differences ($\Delta = \text{inferred} - \text{reference}$). For brevity, “Raw” denotes the baseline strategy, “Non-L1” refers to PCA-Non-L1 ($n = 7$), and “L1” refers to PCA-L1 ($\alpha = 0.06$). Within each emulator–optimizer pair, the minimum σ among the three strategies is highlighted in boldface.

As shown in Figure 11, under the baseline strategy, both model errors and repeat-observation errors generally decrease with increasing S/N. However, when S/N falls below approximately 70, the two diverge significantly: model errors for most parameters are systematically overestimated, such as for T_{eff} , $\log g$, and multiple elemental abundances, where model errors in the low-S/N regime can reach approximately twice the repeat-observation errors. Additionally, certain parameters exhibit extreme error estimates, such as [P/H], [V/H], [Co/H], and [Cu/H], where model errors far exceed repeat-observation errors. This indicates that for spectral emulators, error estimates based on first-order linearization in the neighborhood of the optimal solution may not be sufficiently accurate; particularly when the model exhibits strong nonlinearity with respect to certain parameters, this linearization approximation can introduce biases (K. W. Vugrin et al. 2007). A detailed mechanistic analysis of this phenomenon is beyond the scope of this study. In contrast, the PISP strategy improves this behavior for all parameters except $^{12}\text{C}/^{13}\text{C}$ (see Figure 12): model errors no longer show the extreme discrepancies observed in the baseline strategy (e.g., for [P/H], [V/H], [Co/H], and [Cu/H]), and exhibit stronger consistency with repeat-observation errors in the $S/N < 70$ regime; particularly for $S/N > 70$, the magnitude and trends of model errors and repeat-observation errors are largely consistent. This demonstrates that while improving parameter-inference precision, PISP also enhances the empirical consistency between model errors and repeat-observation errors.

4.6. Optimizer Dependence under Orthogonal Reparameterization

To systematically assess whether PISP’s performance improvement depends on the choice of optimizer, we compare parameter inference results using the FNN emulator across multiple commonly used optimizers on the theoretical spectrum test set. Beyond the default `trf` (T. F. Coleman & Y. Li 1996; M. A. Branch et al. 1999) and Adam (D. P. Kingma & J. Ba 2014) employed in this work, we further test Broyden–Fletcher–Goldfarb–Shanno (BFGS; J. Nocedal & S. J. Wright 2006), limited-memory BFGS (L-BFGS; J. Nocedal & S. J. Wright 2006), Levenberg–Marquardt (`lm`; J. J. Moré 1978), limited-memory BFGS with bound constraints (L-BFGS-B; R. H. Byrd et al. 1995; C. Zhu et al. 1997), `dogleg` algorithm with rectangular trust regions (`dogbox`; C. Voglis & I. E. Lagaris 2004; J. Nocedal & S. J. Wright 2006), and Adam with `ReduceLRonPlateau`. Under ideal conditions where initialization, line search, and stopping criteria are identical and numerical errors are negligible, BFGS and L-BFGS can be regarded as unconstrained methods insensitive to orthogonal reparameterization (hereafter referred to as rotation invariance). Therefore, under the setting of orthogonal reparameterization of standardized stellar parameters using Equation (2) without dimensionality reduction (i.e., PCA-Non-L1 with $n = 25$), their results should theoretically remain essentially identical to the baseline strategy; for these optimizers, we primarily compare $\sigma(\Delta)$ between PCA-L1 ($\alpha = 7 \times 10^{-4}$) and the baseline strategy. In contrast, L-BFGS-B, `lm`, `dogbox`, `trf`, and Adam are typically more sensitive to orthogonal reparameterization (i.e., not rotation invariant) due to

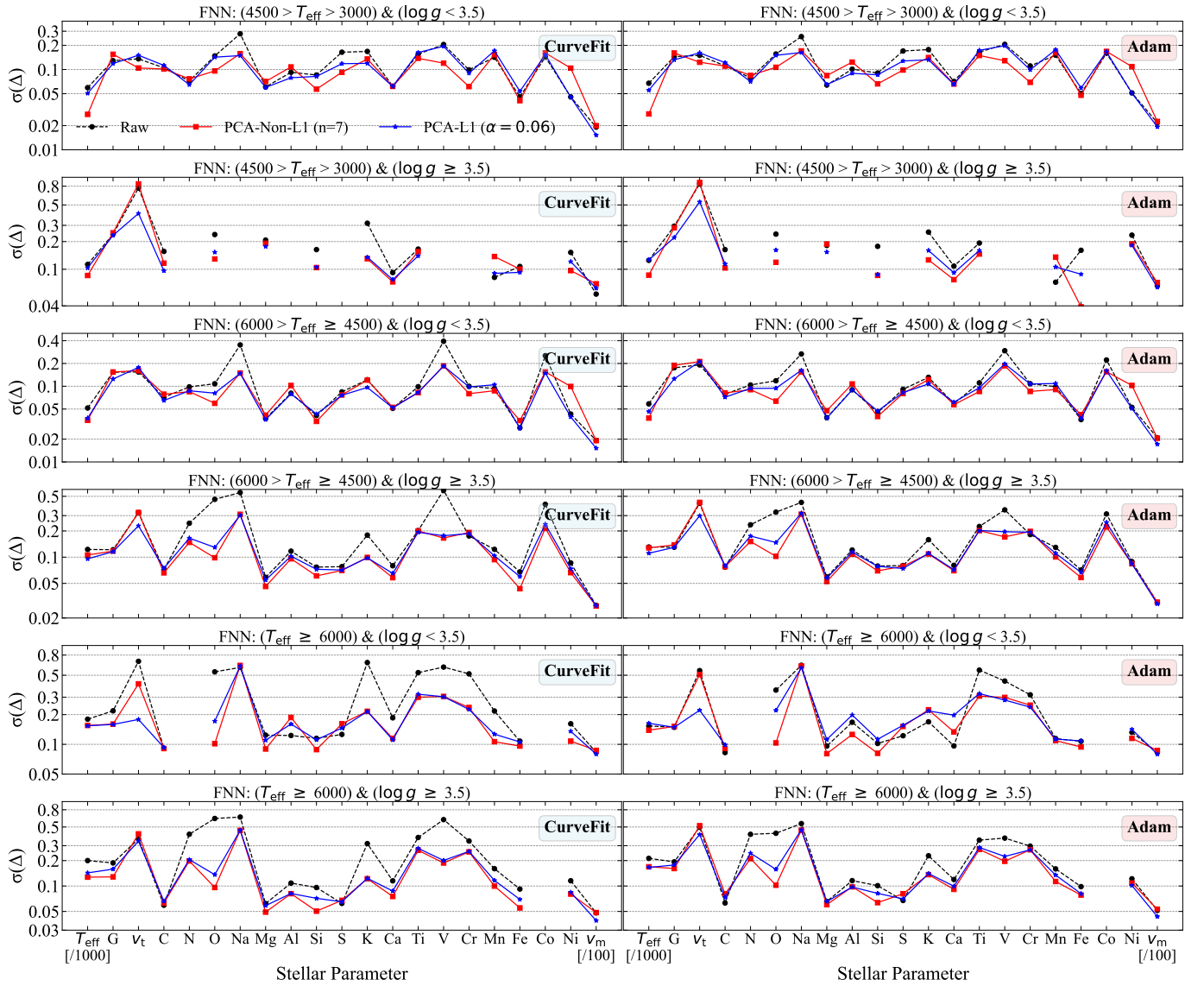


Figure 8. Standard deviations of differences relative to the APOGEE official catalog for PISP and baseline inferences on APOGEE DR17 spectra using the FNN spectral emulator. The 12 panels are organized into six $T_{\text{eff}}\text{-}\log g$ subsets (rows) and the optimizer changes by column (left: CurveFit; right: Adam). In each panel, σ is computed from differences ($\Delta = \text{inferred} - \text{reference}$) for 21 labels using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`). Gaps between curves indicate parameters for which APOGEE catalog values are unavailable after flag-based quality control. For clarity, the legend is shown only in the first panel: the black, red, and blue curves correspond to the baseline strategy described in Section 3.3, the PCA-Non-L1 strategy with $n = 7$, and the PCA-L1 strategy with $\alpha = 0.06$, respectively, each evaluated at its optimal hyperparameter setting. On the x -axis, $\log g$, V_{turb} , and V_{macro} are denoted G , V_t , and V_m ; T_{eff} is shown in units of 10^3 K, and V_m is scaled by 100. The y -axis is base-10 logarithmic with tick labels given in linear units and the dashed horizontal lines indicate reference levels to aid visual comparison across labels.

mechanisms such as bound constraints, trust regions, damping strategies, or per-coordinate adaptive scaling; for these optimizers, we compare both PCA-Non-L1 ($n = 25$) and PCA-L1 ($\alpha = 7 \times 10^{-4}$) against the baseline strategy to characterize the impact of both approaches. Throughout this work, we use the term rotation invariance to refer strictly to the theoretical property under the idealized conditions described in this section; any observed differences under practical implementations should be interpreted as resulting from optimizer-specific mechanisms or numerical errors, rather than a violation of the theoretical property itself.

As shown in Figure 13, PISP’s performance across different optimizer settings can be summarized in three key observations:

1. For rotation-invariant optimizers, PCA-Non-L1 ($n = 25$) is essentially equivalent to the baseline strategy. For BFGS and L-BFGS, PCA-Non-L1 ($n = 25$) yields $\sigma(\Delta)$ closest to that of the baseline strategy, indicating that without dimensionality reduction or L1 regularization, orthogonal reparameterization alone has limited impact on these optimizers. This is because under the transformation defined in Equation (2), the optimization problems corresponding to Equations (3) and (4) are mathematically equivalent: under the aforementioned ideal conditions, iterative updates in the two parameter spaces (standardized stellar-parameter space and basis coefficient space) can be related through Equation (2), yielding nearly identical stellar-parameter inference results when numerical errors are negligible.

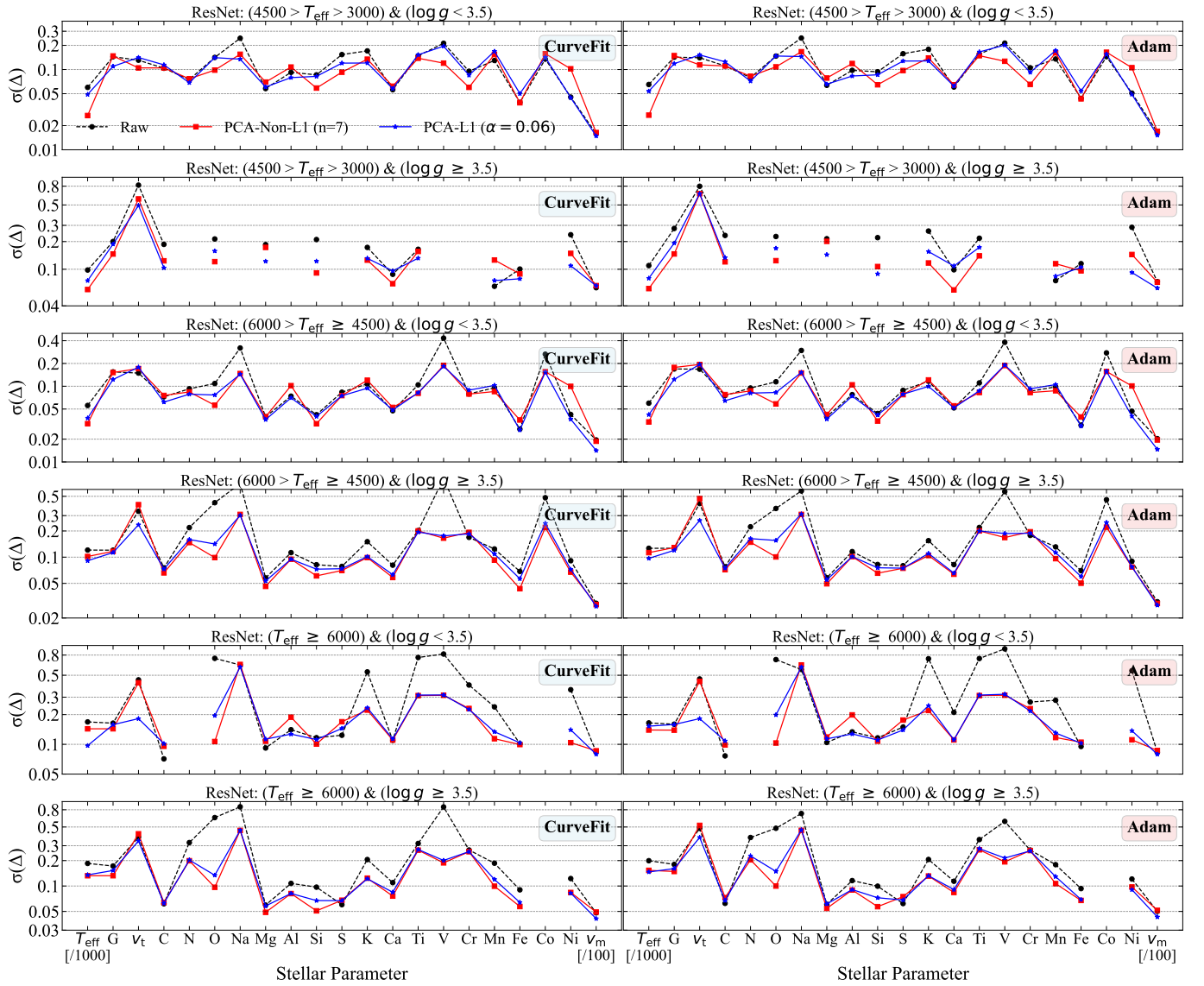


Figure 9. Standard deviations of differences relative to the APOGEE official catalog for PISP and baseline inferences on APOGEE DR17 spectra using the ResNet spectral emulator. Same as Figure 8, but with the FNN emulator replaced by ResNet.

2. For non-rotation-invariant optimizers, PCA-Non-L1 ($n = 25$) outperforms or matches the baseline strategy. In the equal-dimensional, unregularized PCA-Non-L1 ($n = 25$) case, non-rotation-invariant optimizers yield lower or comparable $\sigma(\Delta)$ relative to the baseline strategy. The underlying reason is that for these optimizers, the optimization paths corresponding to Equations (3) and (4) can no longer be kept consistent through Equation (2). In this context, the decorrelation introduced by orthogonal reparameterization serves as a practical and effective data preprocessing technique. Similar ideas appear in regression analysis; for instance, J. Liang et al. (2022) demonstrated that decorrelation processing can improve parameter prediction accuracy while maintaining dimensionality. As a specific example, for Adam under the baseline strategy, introducing ReduceLRonPlateau does not reduce $\sigma(\Delta)$: comparing the RMSE of flux residuals at the optimal solution, we find that Adam with ReduceLRonPlateau reduces RMSE by only $\sim 2 \times 10^{-5}$ relative to Adam

without ReduceLRonPlateau, already below the intrinsic RMSE limit of the spectral emulator (see Figure 1). When optimization approaches this limit, further RMSE reduction does not translate to lower $\sigma(\Delta)$. Moreover, even with ReduceLRonPlateau, the baseline strategy yields higher $\sigma(\Delta)$ than PCA-Non-L1 under the same optimizer.

3. PCA-L1 consistently outperforms across all optimizers. Across all tested optimizers, PCA-L1 achieves the lowest $\sigma(\Delta)$. This reduction is primarily driven by regularization constraints—selecting the regularization strength α in a data-driven manner on the validation or reference set to match data distribution and noise levels. Thus, the PCA-L1 strategy helps reduce $\sigma(\Delta)$ across different optimizer configurations.

Based on these comparisons, under equal-dimensional orthogonal reparameterization on theoretical spectra, PISP's performance trends relative to the baseline strategy are generally consistent across optimizers, indicating that its improvements are primarily driven by reparameterization and

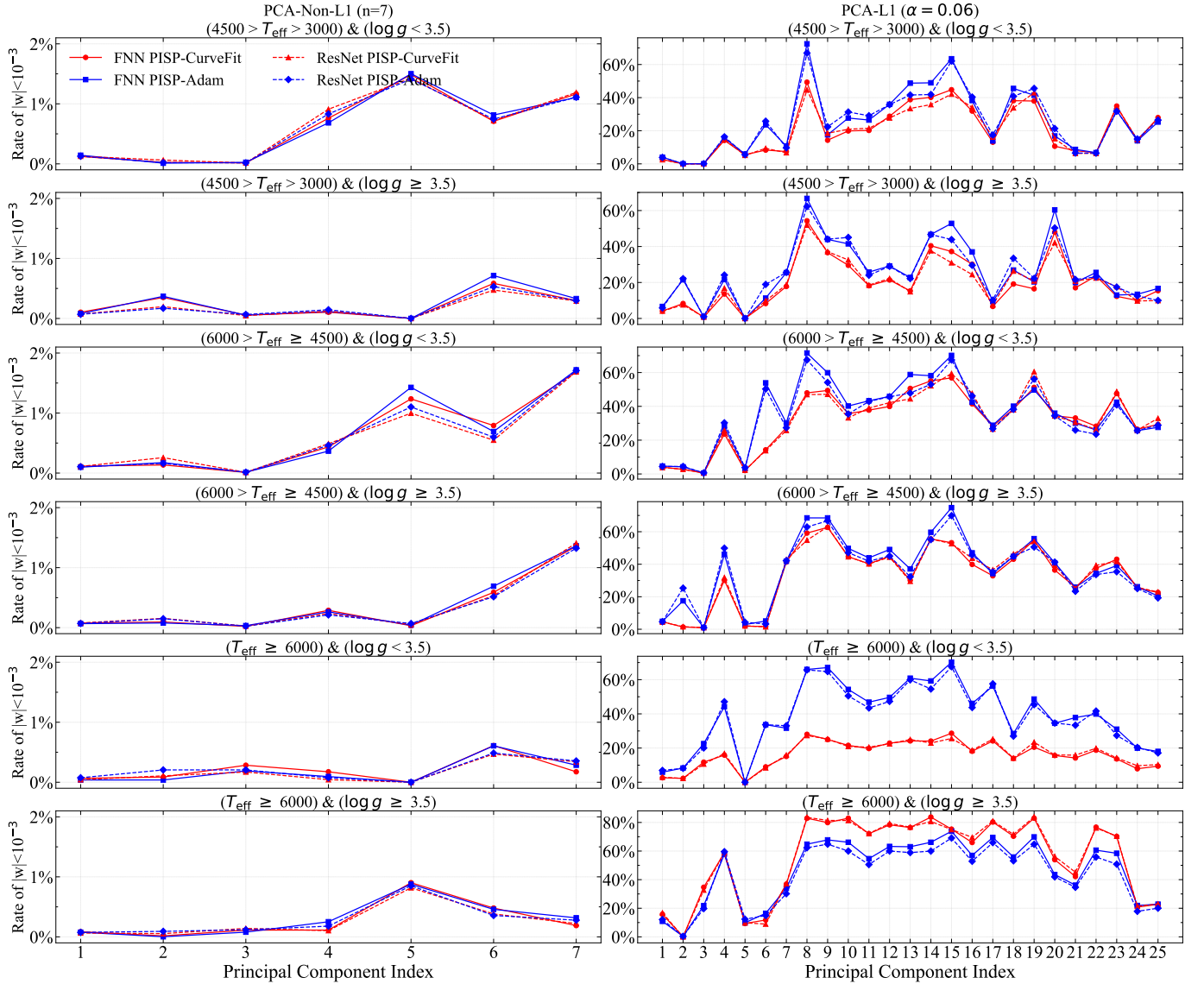


Figure 10. Fractional statistics of principal-component coefficients (w) with absolute values below 10^{-3} derived using PISP on the APOGEE spectra. The first and second columns correspond to the PCA-Non-L1 and PCA-L1 strategies, respectively. The x -axis indicates the index of the principal component and the y -axis shows the fraction of test samples with corresponding coefficients satisfying $|w| < 10^{-3}$. The larger the fraction, the less important the corresponding principal component is for parameter inference. For clarity, the first column shows only the first seven principal components, since the weights of the remaining 18 principal components are 0 (corresponding to 100% on the y -axis). Each panel represents a different $T_{\text{eff}}\text{-}\log g$ subrange. Solid red and blue curves denote results from FNN-based spectral emulators using CurveFit and Adam, respectively, while dashed red and blue curves represent the corresponding results from ResNet-based emulators. For clarity, the legend is shown only in the first panel.

regularization configurations during the inference stage rather than dependence on any specific optimizer setting.

4.7. Efficiency of Stellar Parameter Inference with PISP

To quantify the computational efficiency of PISP for large-scale parameter inference, we evaluated PISP-CurveFit and PISP-Adam on 722,896 observed spectra and compared them with a baseline strategy. All experiments adopted the default n_{jobs} , N , as well as the PCA projection bases and optimal hyperparameters determined in Section 4.2. PISP-CurveFit was run on a blade server node with dual Intel Xeon Gold 5416S processors (32 cores, 64 threads), and PISP-Adam was run on a server equipped with an NVIDIA GeForce RTX 3090.

The same Python virtual environment and dependency versions were used across runs to ensure a fair comparison.

As shown in Table 5, inference efficiency varies substantially across spectral emulator–optimizer–strategy combinations:

1. *Spectral emulator efficiency.* Owing to its lower computational complexity for spectral generation, FNN attains higher throughput than ResNet under the same optimizer and strategy. With the Adam optimizer, this gap is particularly pronounced, and FNN achieves a speedup exceeding $16\times$.
2. *Optimizer efficiency.* Adam supports GPU-accelerated parallel inference and, with an appropriate choice of N (see J.-C. Liang et al. (2026), Figure 1), generally outperforms the CPU-based multithreaded CurveFit (the

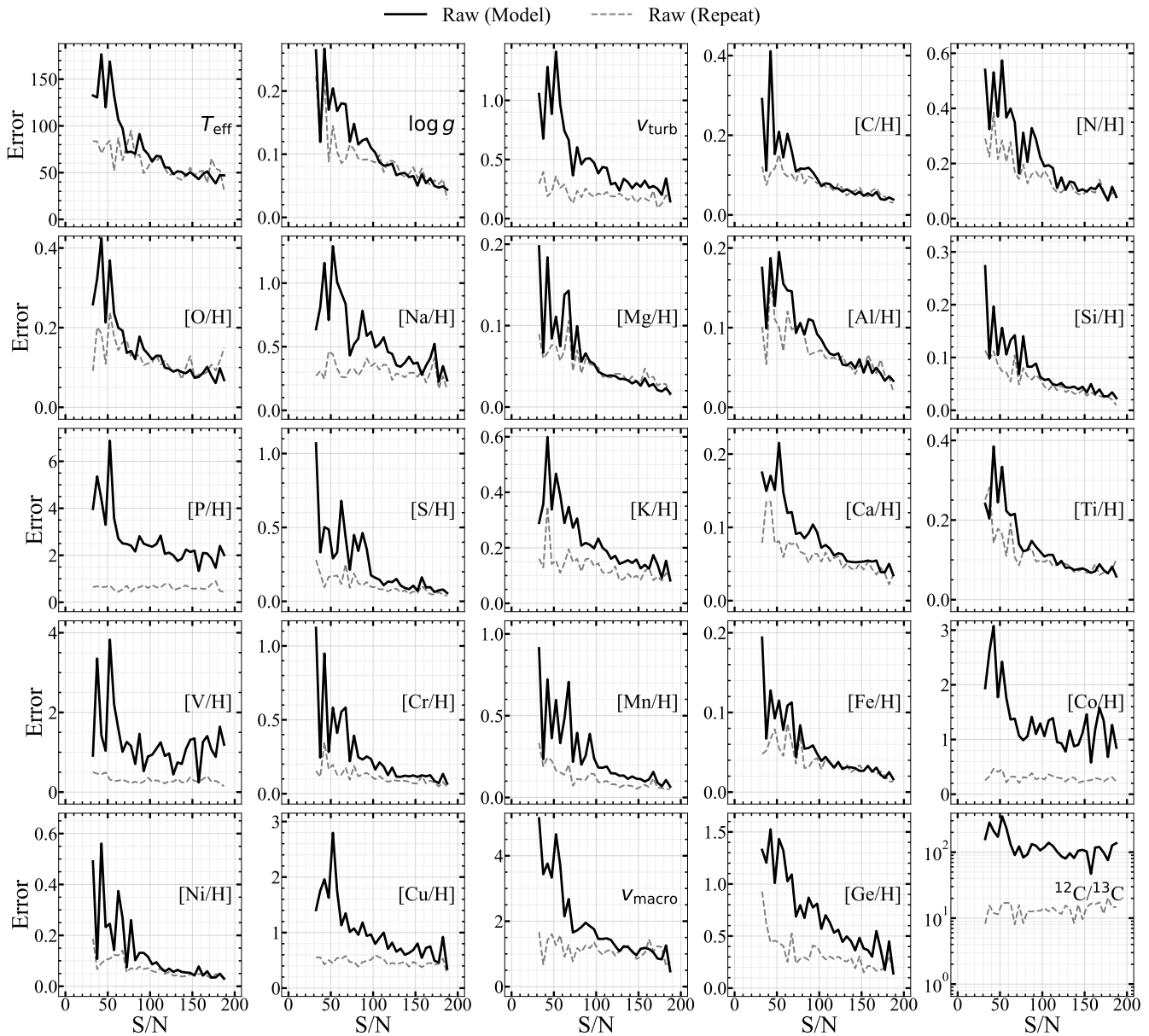


Figure 11. Comparison between model errors (solid curves) and repeat-observation errors (dashed curves) as a function of S/N for the baseline strategy. All results are based on the ResNet spectral emulator and PISP-CurveFit. Errors are computed in S/N bins of width 5 and are reported only for bins containing at least three stars, with the median error shown on the vertical axis. For $^{12}\text{C}/^{13}\text{C}$, the y-axis is base-10 logarithmic with tick labels given in linear units.

ResNet emulator with the PCA-Non-L1 strategy is the exception¹⁴). Using the FNN emulator as an example, Adam yields more than $10\times$ speedup across all three strategies, reaching $\sim 46\times$ under the PCA-L1 strategy (0.49 versus 22.45 hr).

3. *Strategy efficiency.* Relative to the baseline strategy, PCA-Non-L1 reduces the optimization dimensionality from 25 to 7, thereby substantially reducing elapsed runtime (the ResNet + PCA-Non-L1 + Adam combination is an exception), with improvements as high as $\sim 4\times$ (20.78 versus 4.99 hr). In principal-component space,

Adam also requires a substantially smaller number of iterations to reach convergence (see Figure 14); for the FNN emulator, PCA-Non-L1 reduces the mean iteration count by about 242 relative to the baseline strategy. By contrast, the runtime under PCA-L1 changes only marginally, primarily because the decision-variable dimensionality is not reduced a priori and the regularizer introduces additional overhead.

In summary, without sacrificing accuracy, PISP delivers consistent efficiency gains across different emulator–optimizer configurations and is therefore suitable for high-throughput parameter inference on large spectroscopic datasets.

4.8. Future Improvements

To further enhance the inference accuracy of high-dimensional stellar parameters and the adaptability to observed data,

¹⁴ For the ResNet emulator, PISP-CurveFit achieves an efficiency improvement of about a factor of 4 relative to the baseline strategy. In contrast, PISP-Adam requires more iterations to converge (see Figure 14, right panel), leading to slightly lower efficiency. This may be related to the convergence behavior of Adam under the ResNet emulator, while a detailed analysis is beyond the scope of this work.

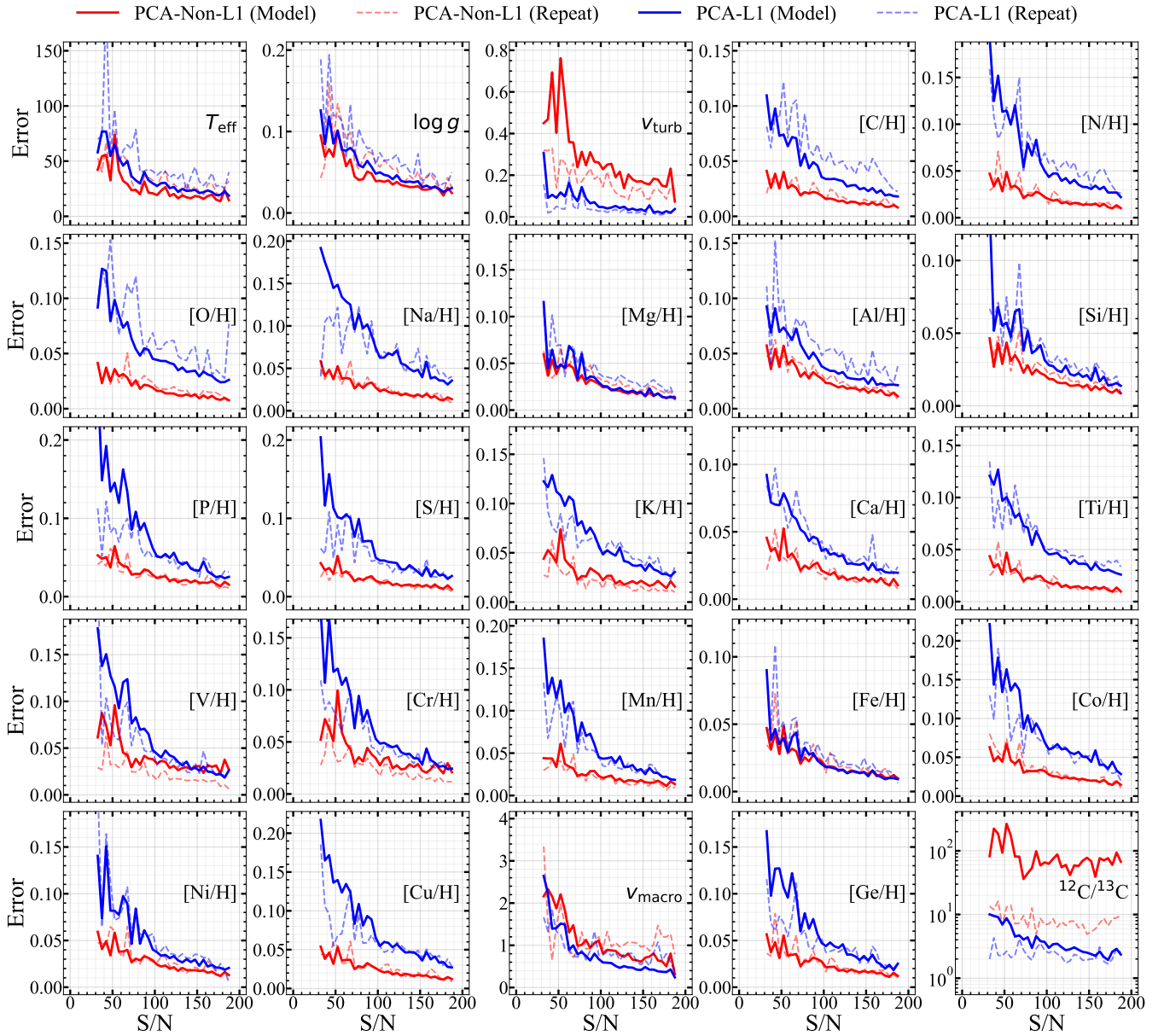


Figure 12. Comparison between model errors (solid curves) and repeat-observation errors (dashed curves) as a function of S/N for PISP. Same as Figure 11, except that the curves are color coded, with red for PCA-Non-L1 ($n = 7$) and blue for PCA-L1 ($\alpha = 0.06$).

Table 5
Inference Time on APOGEE DR17 Spectra Using CurveFit and Adam with Baseline, PCA-Non-L1, and PCA-L1 Strategies

Method	FNN			ResNet		
	Baseline (hr)	PCA-Non-L1 (hr)	PCA-L1 (hr)	Baseline (hr)	PCA-Non-L1 (hr)	PCA-L1 (hr)
CurveFit	20.78	4.99	22.45	27.62	6.59	29.51
Adam	0.50	0.38	0.49	8.37	10.33	7.87

several avenues remain for improving PISP. From the perspectives of parameter-space modeling, adaptation to the observational domain, result reliability, and cross-survey evaluation, we propose the following directions:

1. *Nonlinear embeddings for parameter-space representation.* The current PISP employs linear projection to mitigate the impact of variable correlations on parameter

inference. However, when nonlinear dependencies exist among elemental abundances, its expressiveness is limited. Building on successful applications of nonlinear embeddings in optimization problems (A. Tripp et al. 2020; K. Antonov et al. 2022; N. Maus et al. 2022; J. Chu et al. 2024; O. Boyar & I. Takeuchi 2024), future work can explore kernel principal component analysis and variational autoencoders within the PISP framework

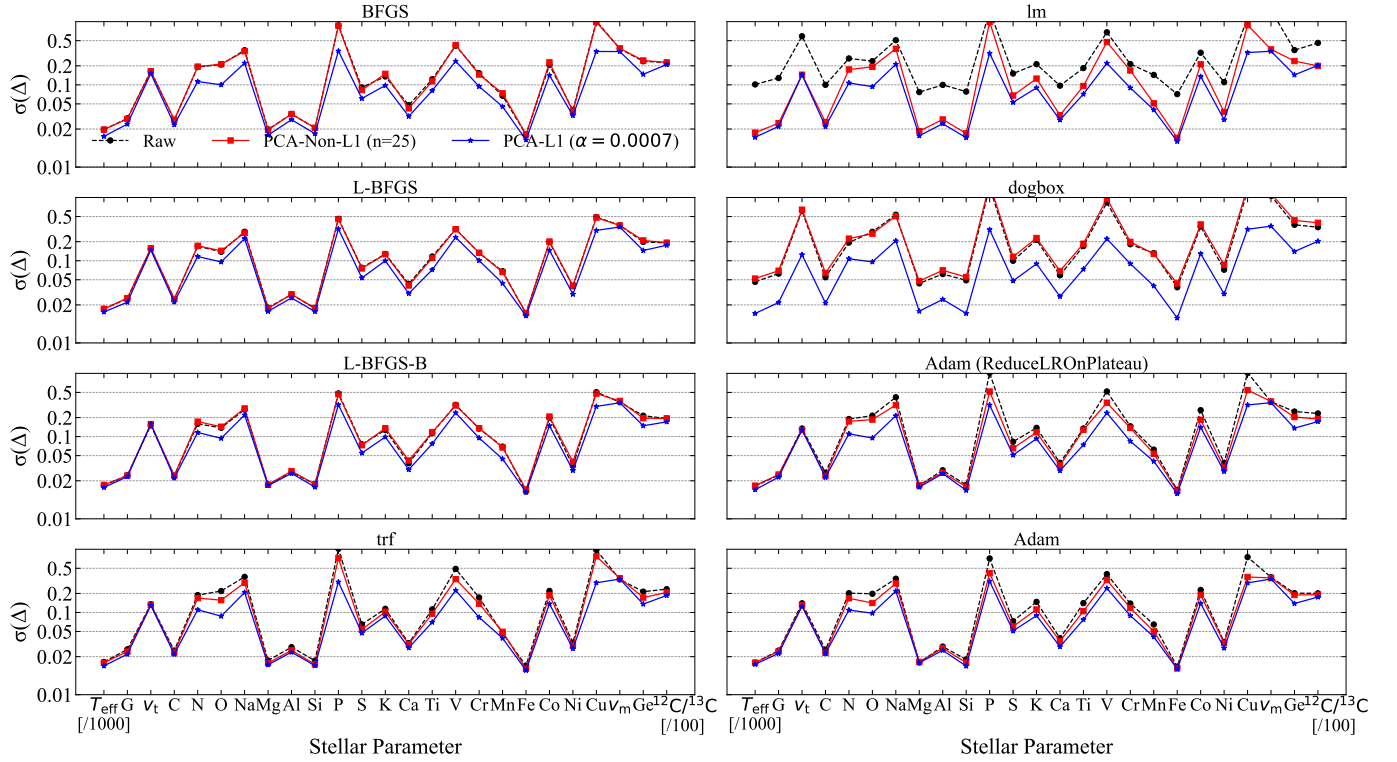


Figure 13. Standard deviations of differences relative to the Kurucz reference labels for PISP and baseline inferences on the test set obtained with different optimizers using the FNN spectral emulator. Each panel corresponds to a different optimizer, including BFGS, L-BFGS, L-BFGS-B, trf, lm, dogbox, Adam with ReduceLRonPlateau, and Adam. In each panel, σ is computed from differences ($\Delta = \text{inferred} - \text{reference}$) for 25 labels using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`). For clarity, the legend is shown only in the first panel: the black, red, and blue curves correspond to the baseline strategy described in Section 3.3, the PCA-Non-L1 strategy with $n = 25$, and the PCA-L1 strategy with $\alpha = 0.0007$, respectively, each evaluated at its optimal hyperparameter setting. On the x -axis, $\log g$, V_{turb} , and V_{macro} are denoted G , V_t , and V_m ; T_{eff} is shown in units of 10^3 K, and $^{12}\text{C}/^{13}\text{C}$ is scaled by 100. The y -axis is base-10 logarithmic with tick labels given in linear units and the dashed horizontal lines indicate reference levels to aid visual comparison across labels.

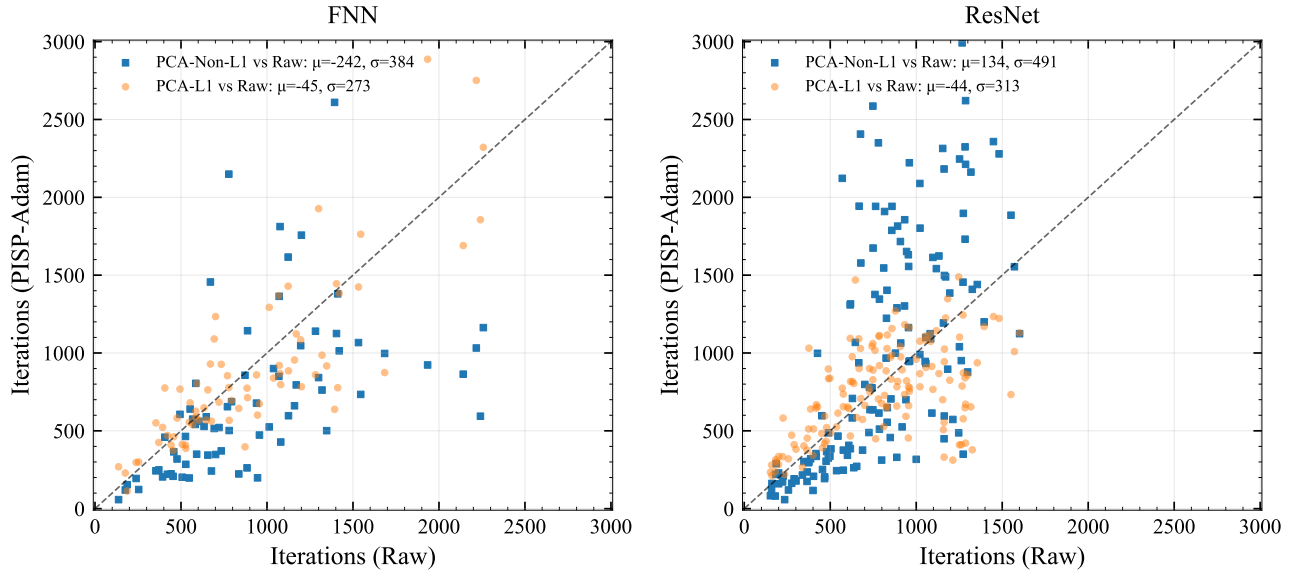


Figure 14. Comparison of iteration counts between PISP-Adam and the baseline strategy for the inference of 25-dimensional stellar parameters from 722,896 APOGEE DR17 spectra. The left and right panels correspond to results using FNN and ResNet spectral emulators, respectively. The left panel includes 73 groups (72 with $N = 10,000$ spectra and one with $N = 2896$), while the right panel includes 145 groups (144 with $N = 5000$ and one with $N = 2896$). In each panel, the μ and σ of differences between the iteration counts of PISP-Adam and those of Adam in the baseline strategy are computed using `astropy.stats.sigma_clipped_stats` (`sigma=3`, `maxiters=3`). The x -axis indicates the number of iterations required by Adam in the baseline strategy and the y -axis shows the iterations required by PISP-Adam in the principal-component space. For PCA-Non-L1, $n = 7$; for PCA-L1, $\alpha = 0.06$.

to strengthen the characterization of nonlinear structure and further improve inference accuracy for certain parameter dimensions.

2. *Observational-domain adaptation and robustness.* Due to distributional differences between synthetic and observed spectra, PISP may underperform on observational data relative to its performance on synthetic test sets. To improve robustness in observations, one may incorporate the *Clean* strategy from the Python-based LAMOST Stellar Parameter Pipeline (PyLASP; J.-C. Liang et al. 2026), which adaptively identifies and removes anomalous pixels with large flux differences during inference, retaining only high-consistency regions for fitting and thereby enhancing noise robustness. In parallel, the Cycle-StarNet method (T. O’Brian et al. 2021) can be leveraged to construct an explicit cross-domain mapping between synthetic and observed spectra, correcting systematic differences at the model level and further improving the inference accuracy of strategies such as PISP on real observations.
3. *Parameter recommendation and per-label quality flags.* Although PISP has produced high-dimensional parameter inferences on large-scale observed spectra, a systematic recommendation mechanism is not yet in place. In practice, some spectra may have low S/Ns or certain parameters may deviate in accuracy or physical consistency due to inherent limitations of both the synthetic spectral models and the inference methodology. To improve the scientific usability of the catalog, we recommend introducing a per-parameter recommendation flag (FLAG) that encodes a credibility level. This mechanism can integrate factors such as spectral quality, objective-function convergence, parameter uncertainty, and distributional plausibility, thereby providing a more robust catalog for subsequent studies. Detailed design and implementation are left for future work.
4. *End-to-end evaluation of official pipelines for high-dimensional stellar-parameter inference.* PISP has not yet undergone systematic evaluation within the official pipelines of large spectroscopic surveys for high-dimensional stellar-parameter inference. Surveys differ substantially in preprocessing, model interfaces, and runtime environments, while the availability and stability of pipelines and training or reference resources also vary. Even where some components are open source, additional engineering integration and recalibration would be required, which is beyond the scope of this study. To ensure fairness and reproducibility, we therefore benchmark PISP against multiple baseline strategies under strictly controlled conditions—using identical data, spectral emulators, and optimization settings. Once interfaces stabilize and integration becomes feasible, we will extend the evaluation to end-to-end validation within native survey pipelines and release the corresponding scripts and results.

5. Conclusions

This work proposes PISP, a projection-space-assisted framework for high-dimensional stellar-parameter inference. By mapping standardized stellar parameters into a projection space via linear transformation, PISP improves both inference

accuracy and efficiency. To accommodate different computing environments and data scales, we implement PISP-CurveFit for fast per-object inference and PISP-Adam for large-scale parallel computation. Using Kurucz synthetic spectra and APOGEE DR17 observed spectra, together with FNN and ResNet spectral emulators, we conduct a systematic evaluation of PISP. The main conclusions are as follows:

1. *Dataset-specific hyperparameter tuning is necessary for optimal performance.* PISP is sensitive to the number of basis vectors n , the L1 regularization strength α , the learning rate lr , and the convergence threshold ϵ . On Kurucz synthetic spectra, a larger n and a smaller α help preserve beneficial directions; on APOGEE observations, a larger α and a smaller n are preferred. Therefore, n and α should be tuned independently according to the task and data distribution. In addition, lr and ϵ primarily affect convergence speed and stability. In this work, we adopt $lr = 0.01$ and $\epsilon = 10^{-8}$ uniformly across datasets.
2. *PISP delivers concurrent gains in accuracy and computational efficiency.* On the Kurucz test set, PCA-L1 performs best, reducing $\sigma(\Delta)$ by at least 0.01 dex for 12 of 20 elemental abundances across all emulator-optimizer combinations, with [N/H], [O/H], [Na/H], [Co/H], [P/H], [V/H], [Cu/H] showing reductions of 0.05–0.72 dex. On APOGEE DR17 observations, PCA-Non-L1 reduces $\sigma(\Delta)$ by more than 30 K for T_{eff} and by at least 0.01 dex for 9 of 17 elemental abundances across all emulator-optimizer combinations, with [O/H], [Na/H], [V/H] showing reductions of 0.05–0.20 dex, exhibiting higher agreement with the official catalog and confirming PISP’s applicability to real survey data. For computational efficiency, PCA-Non-L1 achieves up to $\sim 4\times$ acceleration for large-scale inference on APOGEE data, while PISP-Adam under the PCA-L1 strategy accelerates by up to $46\times$ relative to PISP-CurveFit. These results indicate that PISP balances accuracy and efficiency across datasets and emulator-optimizer combinations, providing a practical solution for high-dimensional stellar-parameter inference in large spectroscopic surveys.
3. *The regularization choice dictates basis-vector selection and guides strategy preference.* L1 and Non-L1 adopt different selection mechanisms: the former induces sparsity in coefficients across all 25 basis vectors via L1 regularization and can dynamically adjust the effective dimensionality for each individual spectrum; the latter fixes a subset of high-variance principal components or active-subspace directions, assuming by default that dominant directions have consistent importance across all samples. Experiments show that PCA-L1 achieves lower $\sigma(\Delta)$ when the training and test distributions are aligned; under distribution shift in observed data, PCA-Non-L1 achieves slightly lower $\sigma(\Delta)$ than PCA-L1, with lower computational cost.
4. *PISP integrates seamlessly across emulators and optimizers without retraining.* Whether using PISP-CurveFit or PISP-Adam, and either the FNN or ResNet spectral emulator, PISP reduces $\sigma(\Delta)$ relative to the baseline strategy without retraining the spectral emulator and thus can be readily integrated into different parameter-inference pipelines.

In summary, PISP demonstrates advantages in accuracy, efficiency, and cross-domain adaptability, providing a general and efficient optimization framework for high-dimensional stellar-parameter inference. In future work, PISP can be integrated with PyLASP and applied to high-dimensional stellar-parameter inference for LAMOST spectra.

Acknowledgments

We thank Yuan-Sen Ting for providing the Kurucz synthetic spectra, without which this work would not have been possible.











This work is supported by the National Natural Science Foundation of China (12273078, 12273075, and 12411530071), the National Key Research and Development Program (2025YFF0510602), and the National Astronomical Observatories of the Chinese Academy of Sciences (No. E4ZR0516). We also acknowledge support from a Royal Society IEC\NSFC\233140 exchange grant.





Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the Participating Institutions.

SDSS-IV acknowledges support and resources from the Center for High Performance Computing at the University of Utah. The SDSS website is www.sdss4.org.

SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration including the Brazilian Participation Group, the Carnegie Institution for Science, Carnegie Mellon University, Center for Astrophysics | Harvard & Smithsonian, the Chilean Participation Group, the French Participation Group, Instituto de Astrofísica de Canarias, The Johns Hopkins University, Kavli Institute for the Physics and Mathematics of the Universe (IPMU)/University of Tokyo, the Korean Participation Group, Lawrence Berkeley National Laboratory, Leibniz Institut für Astrophysik Potsdam (AIP), Max-Planck-Institut für Astronomie (MPIA Heidelberg), Max-Planck-Institut für Astrophysik (MPA Garching), Max-Planck-Institut für Extraterrestrische Physik (MPE), National Astronomical Observatories of China, New Mexico State University, New York University, University of Notre Dame, Observatório Nacional / MCTI, The Ohio State University, Pennsylvania State University, Shanghai Astronomical Observatory, United Kingdom Participation Group, Universidad Nacional Autónoma de México, University of Arizona, University of Colorado Boulder, University of Oxford, University of Portsmouth, University of Utah, University of Virginia, University of Washington, University of Wisconsin, Vanderbilt University, and Yale University.

ORCID iDs

Jun-Chao Liang  <https://orcid.org/0009-0001-9085-8718>
 Yin-Bi Li  <https://orcid.org/0000-0001-7607-2666>
 A-Li Luo  <https://orcid.org/0000-0001-7865-2648>
 Shuo Li  <https://orcid.org/0000-0002-8913-3605>
 Xiao-Xiao Ma  <https://orcid.org/0000-0002-9279-2783>
 Hai-Ling Lu  <https://orcid.org/0000-0002-8252-8743>
 Shu-Guo Ma  <https://orcid.org/0000-0001-5066-5682>
 Ming-Hui Jia  <https://orcid.org/0009-0001-0257-3377>
 Shuo Ye  <https://orcid.org/0009-0005-6200-2778>
 Ke-Fei Wu  <https://orcid.org/0009-0009-0513-7042>

Zhi-Hua Zhong  <https://orcid.org/0009-0003-6636-8203>
 Xiao Kong  <https://orcid.org/0000-0001-8011-8401>
 Li-Li Wang  <https://orcid.org/0000-0001-7783-9662>
 Hugh R. A. Jones  <https://orcid.org/0000-0003-0433-3665>

References

- Abdi, H., & Williams, L. J. 2010, *WIREs Comput. Stat.*, 2, 433
 Abdurro'uf, Accetta, K., Aerts, C., et al. 2022, *ApJS*, 259, 35
 Almeida, A., Anderson, S. F., Argudo-Fernández, M., et al. 2023, *ApJS*, 267, 44
 Antonov, K., Raponi, E., Wang, H., & Doerr, C. 2022, in *Parallel Problem Solving from Nature: PPSN XVII*, ed. G. Rudolph et al. (Springer), 118
 Asplund, M., Grevesse, N., Sauval, A. J., & Scott, P. 2009, *ARA&A*, 47, 481
 Bensby, T., Feltzing, S., & Oey, M. S. 2014, *A&A*, 562, A71
 Blanton, M. R., Bershady, M. A., Abolfathi, B., et al. 2017, *AJ*, 154, 28
 Boyar, O., & Takeuchi, I. 2024, *NComp*, 36, 2446
 Branch, M. A., Coleman, T. F., & Li, Y. 1999, *SJSC*, 21, 1
 Buder, S., Asplund, M., Duong, L., et al. 2018, *MNRAS*, 478, 4513
 Buder, S., Sharma, S., Kos, J., et al. 2021, *MNRAS*, 506, 150
 Buder, S., Kos, J., Wang, X. E., et al. 2025, *PASA*, 42, e051
 Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. 1995, *SJSC*, 16, 1190
 Casey, A. R., Lattanzio, J. C., Aletti, A., et al. 2019, *ApJ*, 887, 73
 Chu, J., Park, J., Lee, S., & Kim, H. J. 2024, in *Advances in Neural Information Processing Systems 37*, ed. A. Globerson et al. (NeurIPS), 68258
 Coleman, T. F., & Li, Y. 1996, *SIAM J. Optim.*, 6, 418
 Constantine, P. G., Dow, E., & Wang, Q. 2014, *SJSC*, 36, A1500
 Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, *RAA*, 12, 1197
 De Silva, G. M., Freeman, K. C., Bland-Hawthorn, J., et al. 2015, *MNRAS*, 449, 2604
 DESI Collaboration, Abdul-Karim, M., Adame, A. G., et al. 2026, *AJ*, 171, 285
 Gaia Collaboration, Vallenari, A., Brown, A. G. A., et al. 2023, *A&A*, 674, A1
 García Pérez, A. E., Allende Prieto, C., Holtzman, J. A., et al. 2016, *AJ*, 151, 144
 Griffith, E. J., Hogg, D. W., Hasselquist, S., et al. 2025, *AJ*, 169, 280
 Griffith, E. J., Weinberg, D. H., Buder, S., et al. 2022, *ApJ*, 931, 23
 Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, 585, 357
 Hotelling, H. 1933, *J. Educ. Psychol.*, 24, 417
 Imig, J., Holtzman, J. A., Yan, R., et al. 2022, *AJ*, 163, 56
 Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
 Letham, B., Calandra, R., Rai, A., & Bakshy, E. 2020, in *Advances in Neural Information Processing Systems 33*, ed. H. Larochelle et al. (NeurIPS), 1546, https://proceedings.neurips.cc/paper_files/paper/2020/file/10fb6cfa4c990d2bad5ddef4f70e8ba2-Paper.pdf
 Liang, J., Bu, Y., Tan, K., et al. 2022, *AJ*, 163, 153
 Liang, J.-C., Li, Y.-B., Luo, A.-L., et al. 2026, *ApJ*, 996, 97
 Luo, A.-L., Zhao, Y.-H., Zhao, G., et al. 2015, *RAA*, 15, 1095
 Majewski, S. R., Schiavon, R. P., Frinchaboy, P. M., et al. 2017, *AJ*, 154, 94
 Martell, S. L., Sharma, S., Buder, S., et al. 2017, *MNRAS*, 465, 3203
 Maus, N., Jones, H., Moore, J., et al. 2022, in *Advances in Neural Information Processing Systems 35*, ed. S. Koyejo et al. (NeurIPS), 34505
 Mead, J., De La Garza Evia, R., & Ness, M. K. 2025, *ApJ*, 994, 52
 Mei, Y., & Wang, H. 2021, arXiv:2105.03687
 Moré, J. J. 1978, *LNM*, 630, 105
 Musayeva, K., & Binois, M. 2024, arXiv:2401.02735
 Ness, M., Hogg, D. W., Rix, H.-W., Ho, A. Y. Q., & Zasowski, G. 2015, *ApJ*, 808, 16
 Ness, M. K., Wheeler, A. J., McKinnon, K., et al. 2022, *ApJ*, 926, 144
 Nissen, P. E., & Gustafsson, B. 2018, *A&ARv*, 26, 6
 Nocedal, J., & Wright, S. J. 2006, *Numerical Optimization* (Springer)
 O'Briain, T., Ting, Y.-S., Fabbro, S., et al. 2021, *ApJ*, 906, 130
 Paszke, A., Gross, S., Massa, F., et al. 2019, in *Advances in Neural Information Processing Systems 32*, ed. H. Wallach et al. (NeurIPS) https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
 Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *JMLR*, 12, 2825
 Piskunov, N., & Valenti, J. A. 2017, *A&A*, 597, A16
 Pretsch, L., Arsenyev, I., Raponi, E., & Duddeck, F. 2024, in *ASME Turbo Expo 2024: Turbomachinery Technical Conf. and Exposition* (American Society of Mechanical Engineers), V12DT34A005
 Ramírez, I., & Allende Prieto, C. 2011, *ApJ*, 743, 135

- Raponi, E., Wang, H., Bujny, M., Boria, S., & Doerr, C. 2020, in *Parallel Problem Solving from Nature: PPSN XVI*, ed. T. Bäck et al. (Springer), 169
- Recio-Blanco, A., de Laverny, P., Palicio, P. A., et al. 2023, *A&A*, **674**, A29
- Rózański, T., Ting, Y.-S., & Jabłońska, M. 2025, *ApJ*, **980**, 66
- Smith, V. V., Bizyaev, D., Cunha, K., et al. 2021, *AJ*, **161**, 254
- Steinmetz, M., Zwitter, T., Siebert, A., et al. 2006, *AJ*, **132**, 1645
- Ting, Y.-S., Conroy, C., Rix, H.-W., & Cargile, P. 2019, *ApJ*, **879**, 69
- Ting, Y.-S., & Weinberg, D. H. 2022, *ApJ*, **927**, 209
- Tripp, A., Daxberger, E., & Hernández-Lobato, J. M. 2020, in *Advances in Neural Information Processing Systems 33*, ed. H. Larochelle et al. (NeurIPS), 11259, https://proceedings.neurips.cc/paper_files/paper/2020/file/81e3225c6ad49623167a4309eb4b2e75-Paper.pdf
- Valenti, J. A., & Piskunov, N. 1996, *A&AS*, **118**, 595
- Varoquaux, G., Grisel, O., Estève, L., et al. 2024, *joblib/joblib*: v1.4.2, Zenodo, doi:10.5281/zenodo.14915602
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *NatMe*, **17**, 261
- Voglis, C., & Lagaris, I. E. 2004, *WSEAS Int. Conf. on Applied Mathematics*, <https://www.wseas.us/e-library/conferences/corfu2004/papers/488-317.pdf>
- Vugrin, K. W., Swiler, L. P., Roberts, R. M., Stucky-Mack, N. J., & Sullivan, S. P. 2007, *WRR*, **43**, W03423
- Wang, Z., Hutter, F., Zoghi, M., Matheson, D., & De Feitas, N. 2016, *J. Artif. Intell. Res.*, **55**, 361
- Weinberg, D. H., Holtzman, J. A., Hasselquist, S., et al. 2019, *ApJ*, **874**, 102
- Weinberg, D. H., Holtzman, J. A., Johnson, J. A., et al. 2022, *ApJS*, **260**, 32
- Xiang, M., Ting, Y.-S., Rix, H.-W., et al. 2019, *ApJS*, **245**, 34
- Yanny, B., Rockosi, C., Newberg, H. J., et al. 2009, *AJ*, **137**, 4377
- Zhang, X., Green, G. M., & Rix, H.-W. 2023, *MNRAS*, **524**, 1855
- Zhao, G., Zhao, Y.-H., Chu, Y.-Q., Jing, Y.-P., & Deng, L.-C. 2012, *RAA*, **12**, 723
- Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. 1997, *ACM Trans. Math. Softw.*, **23**, 550