Draft version March 27, 2026
Typeset using LaTeX **default** style in AASTeX7.0.1

[TODO: Add ORCID for Viska Wei]

SpecViT: A Systematic Comparison of Deep Learning Architectures for Stellar Surface Gravity Estimation from Medium-Resolution Spectra

Viska Wei,[1,2] Xiaosheng Zhao,[1] Rosemary F.G. Wyse,[1] Alexander S. Szalay,[1,2] László Dobos,[1,3] and Tamás Budavári[4,1,2]

[1] *Department of Physics & Astronomy, The Johns Hopkins University, Baltimore, MD 21218, USA*
[2] *Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218, USA*
[3] *Department of Information Systems, Eötvös Loránd University, Budapest 1117, Hungary*
[4] *Department of Applied Mathematics & Statistics, Johns Hopkins University, Baltimore, MD 21218, USA*

Submitted to Astronomy and Computing

ABSTRACT

Accurate inference of stellar atmospheric parameters from noisy spectra is essential for large-scale spectroscopic surveys. While numerous machine learning architectures have been applied to this task, systematic comparisons across architecture families at matched model capacity are rare, making it difficult to attribute performance differences to architecture rather than training choices. We present a comprehensive benchmark of nine deep learning models—spanning convolutional (ResNet-18, Deep CNN), recurrent (BiLSTM), transformer (SpecViT and variants), and hybrid (ConvStem+LSTM, LSTMFormer) architectures—for stellar surface gravity ($\log g$) estimation from medium-resolution one-dimensional spectra. All models use ∼4.8M parameters and are trained on identical synthetic data generated from the BOSZ atmosphere grid with realistic Subaru/PFS noise models (710–885 nm; 4096 pixels).

At matched training data ($N = 2 \times 10^5$), the hybrid ConvStem+LSTM achieves the best performance ($R^2 = 0.729$, $\sigma_{\rm robust} = 0.429$ dex), narrowly surpassing the BiLSTM ($R^2 = 0.728$) and outperforming pure transformers ($R^2 = 0.709$–0.716) and CNNs ($R^2 = 0.640$–0.687). Cross-configuration experiments demonstrate that this ranking is architectural, not an artifact of hyperparameter tuning: transformer models exhibit ∼1000× greater robustness to hyperparameter perturbation than BiLSTM. At higher SNR (mag 19, SNR≈21), the ConvStem+LSTM advantage becomes more pronounced in robust scatter: $\sigma_{\rm robust} = 0.089$ dex versus 0.116 dex for BiLSTM—a 24% reduction in outlier predictions. Using Fisher information analysis, we show that all top models operate within ≈ 0.02 in R^2 of the Cramér–Rao bound at SNR≈4.6, indicating that performance is information-limited rather than model-limited.

Cross-survey validation on APOGEE DR17 demonstrates that transfer from synthetic to real data is *architecture-agnostic*: BiLSTM ($\sigma_{\rm robust} = 0.066$ dex), SpecViT (0.067 dex), and ConvStem+LSTM (0.078 dex) all achieve comparable performance after BOSZ pre-training, outperforming LightGBM (0.111 dex) by 30–40%. We conclude that for spectroscopic stellar parameter estimation: (1) the hybrid ConvStem+LSTM provides the best overall balance of accuracy, robustness, and interpretability; (2) synthetic pre-training is the key enabler of cross-survey transfer, independent of architecture; and (3) architectural innovation alone cannot close the gap to the Fisher ceiling—reducing observational noise has a larger effect.

Corresponding author: Viska Wei

Keywords: Machine learning (1888) — Astronomy data analysis (1858) — Stellar astronomy (1583) — Fundamental parameters of stars (555) — Stellar spectral lines (1630) — Neural networks (1933)

1. INTRODUCTION

Large-scale spectroscopic surveys have made stellar astrophysics a data-rich discipline by delivering millions of spectra with well-characterized instrumental responses and selection functions. Examples range from optical ground-based surveys such as the Sloan Digital Sky Survey (SDSS; D. G. York et al. 2000) and the Large Sky Area Multi-Object Fibre Spectroscopic Telescope (LAMOST; X.-Q. Cui et al. 2012), to space-based missions such as *Gaia* (Gaia Collaboration et al. 2016), and next-generation massively-multiplexed facilities such as DESI (DESI Collaboration 2016) and Subaru/PFS (N. Tamura et al. 2016). From these spectra, a central goal is to infer fundamental stellar parameters—effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity (e.g., [Fe/H])—which are required for stellar population studies, Galactic archaeology, and survey cross-calibration.

Among these labels, surface gravity $\log g$ is particularly important because it separates dwarfs from giants and constrains stellar radii and evolutionary states. Yet $\log g$ is notoriously difficult to estimate robustly from spectra across heterogeneous instruments and signal-to-noise regimes: it is encoded in subtle pressure-broadened wings and line ratios, and is entangled with $T_{\rm eff}$ and composition in ways that depend on wavelength coverage and resolution. Classical pipelines therefore rely on forward modeling and optimization over synthetic libraries, using grids of stellar atmospheres/spectra such as ATLAS9 (F. Castelli & R. L. Kurucz 2004), MARCS (B. Gustafsson et al. 2008), PHOENIX (T.-O. Husser et al. 2013), and BOSZ (R. C. Bohlin et al. 2017). Survey-grade implementations combine these libraries with robust fitting machinery and extensive calibration effort, e.g., the SEGUE Stellar Parameter Pipeline (SSPP; Y. S. Lee et al. 2008), the APOGEE pipeline ASPCAP (A. E. García Pérez et al. 2016), and LAMOST pipelines such as LASP (Y. Wu et al. 2014) and LSP3 (M. Xiang et al. 2015). While physically interpretable, these approaches can be computationally demanding at scale, sensitive to continuum placement and line-spread function mismatch, and limited by modeling systematics when applied to spectra outside the regimes covered by the training/calibration sets.

Data-driven and machine-learning approaches have emerged as powerful alternatives. Methods such as *The Cannon* (M. Ness et al. 2015) learn a generative mapping between spectra and labels, enabling label transfer across surveys, while *The Payne* (Y.-S. Ting et al. 2019) uses neural-network emulators to accelerate spectral synthesis and inference. Discriminative models—including convolutional architectures trained directly on normalized flux sequences—have demonstrated competitive precision for stellar labels on large survey datasets, e.g., StarNet (S. Fabbro et al. 2018) and AstroNN (H. W. Leung & J. Bovy 2019). In parallel, non-neural classical learners such as random forests (L. Breiman 2001) and gradient-boosted decision trees (GBDTs; J. H. Friedman 2001) remain strong baselines in practice, including modern implementations such as LightGBM (G. Ke et al. 2017), and linear regularized models such as ridge regression (A. E. Hoerl & R. W. Kennard 1970). Despite this breadth of methods, a systematic comparison of modern deep learning architecture families—convolutional, recurrent, transformer, and hybrid designs—for stellar spectroscopic regression remains largely absent from the literature. Most studies evaluate a single architecture against classical baselines, making it difficult to disentangle the contribution of architecture choice from training data scale, hyperparameter tuning, and noise augmentation strategy. This gap is particularly consequential because practitioners must choose an architecture before investing in large-scale training, yet no controlled evidence base exists to inform that decision.

Transformers (A. Vaswani et al. 2017) offer a natural mechanism to model global interactions through self-attention. In computer vision, Vision Transformers (ViTs; A. Dosovitskiy et al. 2021) operate on patch tokens and have become high-performing and scalable backbones; later work improved data efficiency and inductive bias (e.g., DeiT H. Touvron et al. 2021, Swin Transformer Z. Liu et al. 2021). For 1D spectra, the same paradigm suggests "patchifying" a spectrum along wavelength and learning a global representation that can attend simultaneously to line cores, wings, and broad-band structure. However, adapting transformers to spectroscopy raises domain-specific questions about tokenization and positional representation: unlike images, spectral pixels have a *physical coordinate* (wavelength) and non-uniform information content. This motivates wavelength-aware position schemes (P. Shaw et al. 2018; J. Su et al. 2021; O. Press et al. 2021) and physics-informed embedding designs. Recent transformer foundation models for stellar spectroscopy—such as SpectraFM (N. Koblischke & J. Bovy 2024)—have demonstrated cross-instrument transfer but face the persistent challenge of bridging the synthetic-to-real domain gap. Controlled benchmarking on synthetic data

therefore remains a necessary first step for validating new architectures before deployment on real observations (e.g., B. Pál et al. 2024).

In this work, we present **SpecViT**, a Vision Transformer adapted for **stellar surface gravity inference** from 1D spectra, and use it as the foundation for a comprehensive architectural comparison. We systematically evaluate convolutional (ResNet-18, Deep CNN), recurrent (BiLSTM), transformer (SpecViT and variants), and hybrid (ConvStem+LSTM, LSTMFormer) architectures at matched parameter budgets on a controlled synthetic benchmark. All models are trained on large-scale synthetic spectra derived from the BOSZ atmosphere library (R. C. Bohlin et al. 2017) with instrument and noise models matched to Subaru/PFS survey conditions. To contextualize performance across signal-to-noise ratios, we additionally derive an information-theoretic ceiling based on Fisher information and the Cramér–Rao lower bound (CRLB; R. A. Fisher 1925; H. Cramér 1946; C. R. Rao 1945; S. M. Kay 1993), providing a principled benchmark for "how close" each architecture is to the best possible estimator under a specified noise model.

Our main contributions are:

- **A systematic architectural benchmark for spectroscopic $\log g$ inference.** We compare nine deep learning models across four architecture families (CNN, RNN, Transformer, Hybrid) at matched parameter budgets ($\sim$4.8M) on 2×10^5 synthetic spectra. The hybrid ConvStem+LSTM ($R^2 = 0.729$) and BiLSTM ($R^2 = 0.728$) lead, followed by pure transformers ($R^2 = 0.709$–0.716) and CNNs ($R^2 = 0.640$–0.687). We demonstrate that this ranking is architectural, not an artifact of hyperparameter tuning (Section 5.1.2). At higher SNR (mag 19), the ConvStem+LSTM advantage in robust scatter becomes more pronounced: $\sigma_{\rm robust} = 0.089$ dex versus 0.116 dex for BiLSTM (24% fewer outlier predictions).

- **Architecture-agnostic cross-survey transfer with synthetic priors.** All deep learning architectures pre-trained on BOSZ synthetic spectra and fine-tuned on 7,000 APOGEE spectra achieve $\sigma_{\rm robust} = 0.066$–$0.078$ dex ($R^2 \geq 0.951$), outperforming LightGBM ($\sigma_{\rm robust} = 0.111$) by 30–40%. Crucially, this transfer succeeds equally for BiLSTM, SpecViT, and ConvStem+LSTM, establishing that the synthetic pre-training strategy—not the architecture—is the key enabler.

- **Near-optimal performance at low SNR and Fisher ceiling analysis.** Using Fisher information analysis, we show that the top deep learning models operate within ≈ 0.02 in R^2 of the theoretical Cramér–Rao bound at SNR$\approx$4.6, demonstrating that performance is information-limited rather than model-limited.

- **Hyperparameter robustness as a practical differentiator.** We show that transformer architectures exhibit $\sim$1000$\times$ greater robustness to hyperparameter perturbation than BiLSTM ($\Delta R^2 \approx 0.007$ vs. > 6), providing a practical advantage in deployment scenarios where extensive tuning is infeasible.

2. RELATED WORK

2.1. *Physics-based stellar parameter inference and survey pipelines*

Stellar parameter inference has traditionally relied on forward modeling with synthetic spectra computed from stellar atmosphere models and radiative transfer. Widely used atmosphere/spectral grids include ATLAS9 (F. Castelli & R. L. Kurucz 2004), MARCS (B. Gustafsson et al. 2008), PHOENIX (T.-O. Husser et al. 2013), and BOSZ (R. C. Bohlin et al. 2017). To fit observations with such grids, the community developed robust optimization and synthesis tools such as SME (J. A. Valenti & N. Piskunov 1996), iSpec (S. Blanco-Cuaresma et al. 2014), and FERRE (C. Allende Prieto et al. 2006; C. Allende Prieto 2016). Large surveys operationalized these ideas in automated pipelines, e.g., SSPP for SDSS/SEGUE (Y. S. Lee et al. 2008), ASPCAP for APOGEE (A. E. García Pérez et al. 2016), and LAMOST pipelines such as LASP (Y. Wu et al. 2014) and LSP3 (M. Xiang et al. 2015). These pipelines achieve strong accuracy and interpretability but often require careful continuum normalization, line-spread-function calibration, and extensive post-hoc corrections; they may also be bottlenecked by repeated grid evaluation and by model mismatch when physics is incomplete or when instrument systematics are not fully captured. A central open question is whether learned representations can match or exceed physics-based methods while scaling to the data volumes anticipated by next-generation surveys.

2.2. *Data-driven models and machine learning on stellar spectra*

Data-driven modeling can reduce reliance on explicit spectral synthesis while retaining physical supervision through labels. *The Cannon* (M. Ness et al. 2015) learns a generative relationship between flux and labels and has been

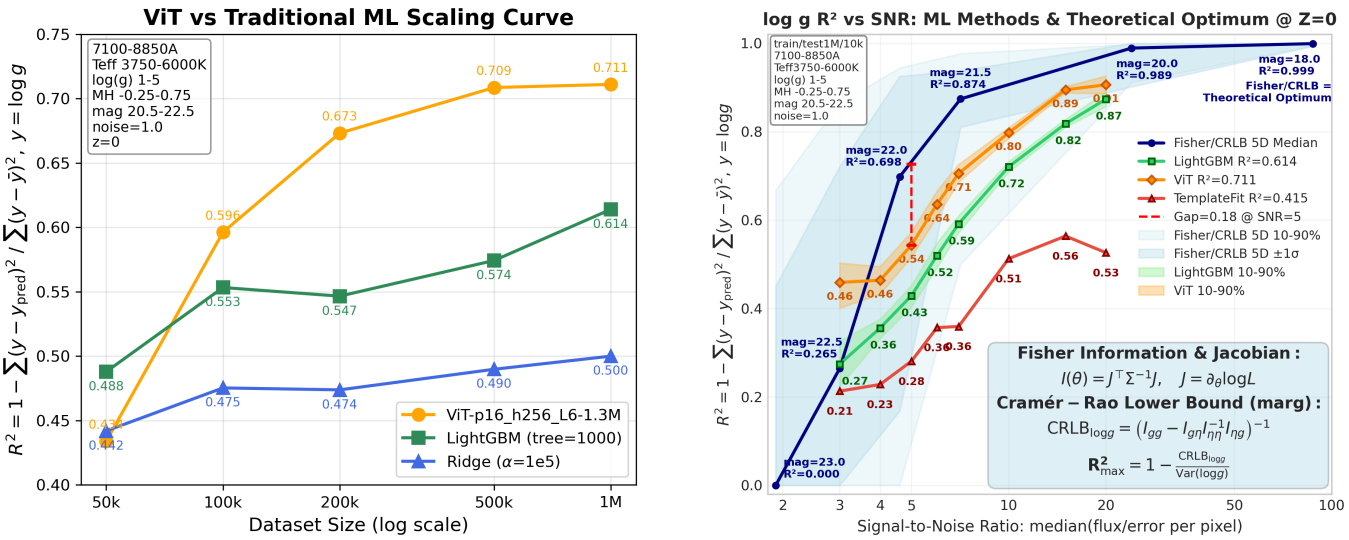

Figure 1. **Left:** Scaling of $\log g$ inference performance with training set size N. Deep learning models improve rapidly with data and overtake LightGBM at $N \sim 10^5$, then saturate beyond $N \gtrsim 5 \times 10^5$, suggesting a transition to a data-diversity-limited regime. **Right:** Test-set R^2 for $\log g$ inference as a function of signal-to-noise ratio (SNR). The top deep learning models remain the best-performing learned models across SNR and approach the Fisher-information–based theoretical ceiling (dashed) at low-SNR conditions.

used for label transfer and homogenization across datasets. *The Payne* (Y.-S. Ting et al. 2019) accelerates forward modeling by learning a neural approximation to spectral synthesis, enabling rapid inference and facilitating large-scale abundance studies. Complementary discriminative deep-learning approaches train directly on spectra to regress stellar labels. CNN-based models such as StarNet (S. Fabbro et al. 2018) and AstroNN (H. W. Leung & J. Bovy 2019) have demonstrated strong performance on APOGEE-like spectra, especially when trained on large, consistently processed datasets. Residual networks have also been applied: Z. Li et al. (2025) show that a fully connected ResNet achieves competitive accuracy with minimal model size on LAMOST spectra. Recurrent architectures such as GRU networks (StarGRUNet; J. Li et al. 2023) provide an alternative that captures sequential dependencies in spectra. In practical survey settings, "classical" machine learning methods remain competitive, including random forests (L. Breiman 2001), support vector machines (C. Cortes & V. Vapnik 1995), and gradient boosting (J. H. Friedman 2001) implemented efficiently in LightGBM (G. Ke et al. 2017). Linear baselines such as ridge regression (A. E. Hoerl & R. W. Kennard 1970) are also attractive when interpretability, stability, or calibration is prioritized.

A recurring challenge for data-driven models is generalization beyond the training distribution: changes in resolution, wavelength range, line-spread function, or continuum normalization can all produce significant performance degradation. This motivates architectures that can ingest variable-length inputs and that can represent spectra in a way that is less tied to a single instrument.

2.3. *Transformers, ViTs, and positional representations for 1D signals*

Transformers (A. Vaswani et al. 2017) replaced recurrence with self-attention and have become a general-purpose architecture for sequence modeling. Vision Transformers (A. Dosovitskiy et al. 2021) extended this paradigm to images by operating on patch tokens, with subsequent refinements that improve data efficiency and inductive bias (e.g., DeiT H. Touvron et al. 2021, Swin Z. Liu et al. 2021). A key design choice for transformers is how to inject positional information. Absolute sinusoidal embeddings were introduced with the original transformer (A. Vaswani et al. 2017), while later work proposed relative-position mechanisms (P. Shaw et al. 2018) and extrapolation-friendly biases such as ALiBi (O. Press et al. 2021). Rotary position embeddings (RoPE) provide another effective mechanism that mixes absolute and relative position information in attention (J. Su et al. 2021). These advances are directly relevant when transferring transformers to spectroscopy, where the physical coordinate (wavelength) and the locality of spectral features create structure that is not identical to either text or images. However, no prior work has systematically compared how these positional strategies interact with the sequential and local inductive biases provided by recurrent and convolutional architectures on spectroscopic regression tasks.

2.4. *Transformers for spectroscopy and astronomy*

Transformer models are increasingly explored in astronomy for heterogeneous modalities. For stellar spectroscopy specifically, recent work has investigated pre-training and transfer learning with transformer architectures. SpectraFM (N. Koblischke & J. Bovy 2024) proposes a transformer "foundation model" for stellar spectra with per-pixel wavelength+flux tokenization, demonstrating cross-instrument transfer but limited to 512 pixels by computational constraints. SpecTE (X. Zhao et al. 2025) achieves state-of-the-art performance on LAMOST low-resolution spectra ($\log g$ MAE= 0.08 dex at SNR$\geq$ 5) using a denoising pretraining strategy that maps noisy observations to clean spectra before supervised fine-tuning, processing 9.8 million LAMOST DR11 spectra at production scale. OmniSpectra (M. K. Islam & J. Fox 2026) pursues a unified multi-survey foundation model with adaptive patching across variable wavelengths, enabling native-resolution learning from multiple surveys simultaneously. In extragalactic spectroscopy, SpecPT (R. Pattnaik et al. 2025) uses a pre-trained transformer for spectrum reconstruction and automated redshift inference on DESI-like data. TransformerPayne (T. Rozański et al. 2025) applies transformers to spectral emulation (the label→spectrum direction), demonstrating superior transfer learning with 2–5× lower flux error than scaled-up MLPs, and recent work on scaling laws for spectral emulators (T. Rozański & Y.-S. Ting 2025) shows that NLP-style power-law relationships extend to this domain. Other studies have explored applying vision transformer pipelines to spectral analysis by representing spectra as images and using ViT backbones (C. M. Moraes et al. 2025), and related transformer applications appear in adjacent problems such as stellar classification using SDSS photometric images (Y. Yang & X. Li 2024). Compared with these efforts, our focus is on providing a *systematic comparison across architecture families*—including recurrent and hybrid designs that have received less attention—for high-accuracy regression of *stellar surface gravity* from 1D spectra at the faint end of next-generation surveys (mag 20.5–22.5, SNR≈3–24), with an emphasis on scaling behavior, hyperparameter robustness, and benchmarking against an information-theoretic limit. We note that direct comparison with SpecTE's 0.08 dex performance is not straightforward: SpecTE operates on LAMOST optical spectra ($R \approx 1800$, SNR$\geq$ 5), while our benchmark targets PFS near-infrared simulations ($R \approx 5000$) at significantly fainter magnitudes where the noise-dominated regime poses fundamentally different challenges. A related open question concerns data scaling: while neural scaling laws (J. Kaplan et al. 2020; J. Hoffmann et al. 2022) have established predictable performance–data–capacity relationships for language models, analogous scaling behavior for scientific regression tasks—particularly spectroscopy—remains unexplored, leaving practitioners without guidance on how much training data different architectures require.

2.5. *Physics-informed learning and information-theoretic limits*

Physics-informed machine learning aims to incorporate physical structure, constraints, or priors to improve sample efficiency and generalization. Physics-informed neural networks (PINNs) (M. Raissi et al. 2019) and broader reviews of physics-informed ML (G. E. Karniadakis et al. 2021) highlight strategies for integrating known laws or inductive biases into learning systems. In stellar spectroscopy, one concrete "physics" is the wavelength coordinate itself and the expected locality and line-formation structure, motivating wavelength-aware embeddings and tokenization schemes (e.g., N. Koblischke & J. Bovy 2024).

Finally, the Fisher information and the Cramér–Rao lower bound provide a principled way to quantify the best attainable variance of unbiased estimators under a specified likelihood (R. A. Fisher 1925; H. Cramér 1946; C. R. Rao 1945; S. M. Kay 1993). Fisher-matrix analyses are widely used in astrophysics to forecast parameter constraints (e.g., M. Tegmark et al. 1997). In this paper, we use Fisher/CRLB analysis not as a cosmological forecast tool but as a *task-level performance ceiling* for $\log g$ inference under our spectral noise model, enabling an interpretable comparison between deep learning architectures, classical machine-learning baselines, and the theoretical limit.

3. METHOD

3.1. *Overview of the Approach*

Figure 2 illustrates the complete pipeline. The goal is to infer the stellar surface gravity, $\log g$, from a medium-resolution, one-dimensional spectrum observed on a fixed detector wavelength grid. Let $\boldsymbol{\lambda} = \lambda_j{}_{j=1}^{L}$ denote the wavelength sampling and $\boldsymbol{f} \in \mathbb{R}^L$ the corresponding (continuum-normalized) flux vector, with $L \simeq 4 \times 10^3$ pixels. The target is a continuous scalar $y \equiv \log g$.

SpecViT models the conditional mapping

$$\hat{y} = F_{\boldsymbol{\theta}}(\boldsymbol{f}). \tag{1}$$

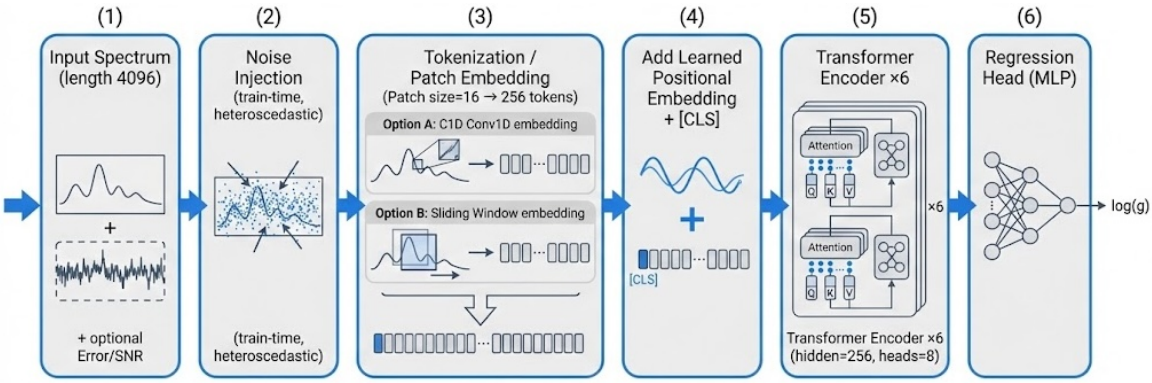

Figure 2. Overview of the SPECVIT pipeline. A 1D stellar spectrum is partitioned into contiguous wavelength patches, each embedded into a latent token. The token sequence, augmented with a learnable summary token and positional embeddings, is processed by a Transformer encoder stack. The final summary token representation is passed to a regression head to predict $\log g$.

where $F_{\boldsymbol{\theta}}$ is a Transformer encoder operating on a sequence of learned "spectral tokens" constructed from contiguous wavelength patches. This design reflects two properties of stellar spectra that are central to $\log g$ inference: (i) *local* information encoded in line cores and pressure-broadened wings, and (ii) *global* constraints arising from line blending, the coupling of multiple diagnostics across the bandpass, and the need to down-weight noise-dominated regions. A self-attention encoder provides a statistically flexible mechanism to aggregate information across widely separated wavelength regions while maintaining an explicit wavelength ordering through positional embeddings.

Traditional forward modeling approaches interpret spectra via a generative function $f(\boldsymbol{\phi})$ evaluated on a grid of stellar parameters $\boldsymbol{\phi}$, often combined with χ^2 minimization under a noise model. In contrast, SPECVIT is a discriminative estimator trained to approximate the optimal regression functional under the same observational noise assumptions, while preserving the key physical structure that the input is a *wavelength-ordered* signal with localized spectral features.

3.2. *Spectral Representation and Preprocessing*

Each spectrum is provided on a fixed detector grid as $(\boldsymbol{\lambda}, \boldsymbol{f}, \boldsymbol{\sigma})$, where $\boldsymbol{\sigma} \in \mathbb{R}^L$ is the per-pixel 1σ uncertainty (heteroscedastic across wavelength). The spectra are generated from synthetic stellar atmosphere models and processed to a medium-resolution instrumental configuration, after which they are sampled onto the common wavelength grid.

Flux normalization.—A robust continuum scaling is required because absolute flux levels depend on distance, throughput, and observing conditions, whereas $\log g$ is encoded primarily in the *relative* shapes and depths of spectral features. A per-spectrum median normalization is adopted,

$$x_j \equiv \frac{f_j}{\mathrm{median}(\boldsymbol{f})}. \tag{2}$$

yielding a dimensionless input vector $\boldsymbol{x} \in \mathbb{R}^L$. In practice, negative flux excursions can arise from noise in low-SNR regimes; since the physically expected photon flux is non-negative, values are clipped at zero after normalization to avoid introducing unphysical large-magnitude outliers that disproportionately affect the regression loss.

Noise model and uncertainty handling.—Observational noise is modeled as independent Gaussian perturbations with known per-pixel variance,

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \alpha(\boldsymbol{\epsilon} \odot \boldsymbol{\sigma}), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

where $\odot$ denotes element-wise multiplication and α is a noise-level scaling factor (unity corresponds to the nominal instrument-model uncertainties). This corresponds to a diagonal noise covariance $\boldsymbol{\Sigma} = \mathrm{diag}(\alpha^2\sigma_1^2, \ldots, \alpha^2\sigma_L^2)$. The uncertainty vector $\boldsymbol{\sigma}$ is therefore used to generate physically consistent noise realizations and to expose the model during training to the same heteroscedastic structure expected at inference time.

3.3. *Patch-based Tokenization of 1D Spectra*

The normalized spectrum $\tilde{\boldsymbol{x}} \in \mathbb{R}^L$ is converted into a sequence of tokens by partitioning the wavelength axis into contiguous patches. For a patch length P and stride S, the i-th patch is

$$\tilde{\boldsymbol{x}}^{(i)} = \left(\tilde{x}_{(i-1)S+1}, \ldots, \tilde{x}_{(i-1)S+P}\right) \in \mathbb{R}^P, \qquad i = 1, \ldots, N. \tag{4}$$

with $N = \left\lfloor \frac{L-P}{S} \right\rfloor + 1$. In the default non-overlapping case $S = P$, and for $L = 4096$ and $P = 16$, this yields $N = 256$ tokens.

Patch embedding.—Each patch is mapped into a D-dimensional latent space through a learnable projection,

$$\boldsymbol{e}_i = \mathcal{E}\left(\tilde{\boldsymbol{x}}^{(i)}\right) \in \mathbb{R}^D. \tag{5}$$

Operationally, $\mathcal{E}$ can be implemented as a one-dimensional convolution with kernel size P and stride S, which shares parameters across wavelength and acts as a bank of learned local filters. This imposes a mild locality prior appropriate for spectroscopy: narrow spectral regions carry coherent information (e.g., line profiles and blends) while still allowing the subsequent attention layers to couple distant regions. An alternative "sliding window" implementation using `unfold()` followed by a linear projection was also explored; however, this configuration exhibited training instability due to larger gradient magnitudes in the Transformer layers (approximately 2× compared to the convolutional approach), leading to model collapse in practice. Appendix B.2 provides further details on this comparison. All results reported in this paper use the convolutional (Conv1D) patch embedding unless otherwise noted.

Motivation.—Tokenizing by wavelength patches balances information retention and computational tractability. Pixel-wise tokenization preserves maximal detail but yields long sequences and encourages the model to spend capacity on high-frequency noise. Conversely, line-list-based or hand-engineered representations can be physically motivated but require strong prior choices about which diagnostics matter for $\log g$ and how blends should be treated. Patch tokens provide a uniform, instrument-agnostic representation that remains aligned with the data acquisition process (detector pixels), while allowing the model to learn which regions are informative under varying SNR.

3.4. *Transformer Encoder Architecture*

Input sequence construction.—A learnable global token (analogous to a summary token) is prepended to the patch embeddings and absolute positional embeddings are added to preserve wavelength order:

$$\mathbf{Z}_0 = [\boldsymbol{z}_{\text{cls}}; \boldsymbol{e}_1; \ldots; \boldsymbol{e}_N] + \mathbf{P}, \qquad \mathbf{Z}_0 \in \mathbb{R}^{(N+1)\times D}, \tag{6}$$

where $\boldsymbol{z}_{\text{cls}} \in \mathbb{R}^D$ is a learned vector and $\mathbf{P} \in \mathbb{R}^{(N+1)\times D}$ encodes the token positions along the wavelength axis. Because spectral features are tied to specific rest-frame wavelengths, absolute positional information is essential; without it, a model with translation symmetry would be poorly matched to spectroscopy.

Self-attention.—The sequence is processed by a stack of L_{enc} Transformer encoder layers. Given an input $\mathbf{Z} \in \mathbb{R}^{(N+1)\times D}$, a single attention head computes

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{Z}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{Z}\mathbf{W}^V, \tag{7}$$

followed by

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_k}}\right)\mathbf{V}. \tag{8}$$

where d_k is the per-head key dimension. Multi-head self-attention concatenates the outputs of H heads and applies an output projection:

$$\text{MSA}(\mathbf{Z}) = \text{Concat}\left(\text{Attn}_1, \ldots, \text{Attn}_H\right)\mathbf{W}^O. \tag{9}$$

In the context of spectroscopy, Eq. (8) allows the model to learn data-adaptive couplings between wavelength regions, such as correlations between pressure-sensitive line wings and metallicity-sensitive blends elsewhere in the spectrum, while naturally down-weighting pixels whose information content is suppressed by noise.

Encoder block.—Each layer uses residual connections and layer normalization in a pre-normalization form:

$$\mathbf{Z}'_\ell = \mathbf{Z}_{\ell-1} + \mathrm{MSA}\left(\mathrm{LN}(\mathbf{Z}_{\ell-1})\right), \tag{10}$$

$$\mathbf{Z}_\ell = \mathbf{Z}'_\ell + \mathrm{MLP}\left(\mathrm{LN}(\mathbf{Z}'_\ell)\right), \qquad \ell = 1, \ldots, L_{\mathrm{enc}}. \tag{11}$$

where MLP is a position-wise feed-forward network. The architecture is parameterized by (D, H, L_{enc}); the configuration used in the main experiments is $D = 256$, $H = 8$, and $L_{\mathrm{enc}} = 6$ (4.8M parameters), which provides sufficient capacity for medium-resolution spectra while keeping the token sequence length moderate ($N_{\mathrm{tok}} = 4096/16 + 1 = 257$).

3.5. *Hybrid Architecture: ConvStem+LSTM*

A key finding of our benchmark (Section 5.1) is that models incorporating recurrent layers consistently outperform pure transformers for spectroscopic regression at moderate data scales. Motivated by this observation, we design the CONVSTEM+LSTM hybrid that combines the strengths of convolutional, recurrent, and transformer components.

Architecture.—The CONVSTEM+LSTM replaces the single Conv1D tokenizer with a three-stage embedding pipeline:

1. **ConvStem tokenizer** (130K parameters): A 4-layer CNN with progressively increasing channel depth ($1 \to 32 \to 64 \to 128 \to 256$) and combined stride of 16, providing a 31-pixel receptive field. This captures complete absorption line profiles (Ca II triplet lines span 35–47 pixels across 2–3 patches with the standard Conv1D tokenizer, but are fully resolved within the ConvStem's receptive field).

2. **BiLSTM context enrichment** (395K parameters): A single-layer bidirectional LSTM ($h = 128$) processes the patch sequence, enriching each patch representation with full bidirectional context. This addresses a key limitation of patch-based tokenization: the Conv1D tokenizer projects each patch independently, with no cross-patch communication at the tokenization stage. The LSTM output is projected back to D dimensions and combined with the original patch embeddings via a residual connection and layer normalization:

$$\boldsymbol{e}'_i = \mathrm{LN}\left(\boldsymbol{e}_i + \mathrm{Linear}\left(\mathrm{BiLSTM}(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N)_i\right)\right). \tag{12}$$

   The residual connection and layer normalization at the LSTM→Transformer boundary are critical for training stability; without them, the hybrid model fails to converge (see Section 5.1.2).

3. **Transformer encoder** (4.09M parameters): A 5-layer transformer encoder (reduced from 6 layers to accommodate the LSTM parameter budget) with the standard CLS token readout and positional embeddings, as described in Section 3.4.

The total parameter count is 4.61M, within the ∼4.8M budget shared by all models. The key design principle is that the BiLSTM provides immediate bidirectional context *before* the transformer layers, enriching each patch token with information from its neighbors—a capability that pure transformers must learn from data through multiple attention layers.

3.6. *Baseline Architectures*

To provide a comprehensive comparison, we implement parameter-matched baselines across architecture families:

BiLSTM (4.2M parameters).—A 3-layer bidirectional LSTM that shares the same Conv1D patch tokenizer as SPECVIT. The spectrum is tokenized into 256 patch embeddings, processed by the BiLSTM stack, and the final hidden states (concatenated forward and backward) are passed through a linear regression head. This architecture provides the strongest single-family baseline due to its native sequential inductive bias.

ResNet-18 (4.9M parameters).—A 1D adaptation of the ResNet-18 architecture (K. He et al. 2016) with residual blocks and channel widths [72, 144, 288, 576]. The raw spectrum is processed by stacked convolutional blocks with skip connections, followed by global average pooling and a linear head.

Deep CNN (4.7M parameters).—An 8-layer deep CNN extending the StarNet architecture (S. Fabbro et al. 2018), with batch normalization, ReLU activations, and progressive max-pooling. This represents a more traditional CNN approach without skip connections.

LSTMFormer (4.8M parameters).—A variant of the hybrid design using a standard Conv1D tokenizer (instead of ConvStem) with a 1-layer BiLSTM ($h = 192$) followed by a 5-layer transformer encoder. This ablates the contribution of the ConvStem tokenizer.

OverlapViT (4.8M parameters).—A pure transformer with overlapping patches (kernel size 32, stride 16), doubling the Conv1D receptive field so that spectral lines at patch boundaries are fully represented in at least one patch.

3.7. *Regression Head for Stellar Parameter Inference*

The model prediction is derived from the encoded summary token. Let $\boldsymbol{h} \equiv \mathbf{Z}_{L_{\text{enc}}}[0] \in \mathbb{R}^D$ denote the final-layer embedding of the prepended token. A regression head maps $\boldsymbol{h}$ to $\hat{y}$:

$$\hat{y} = g(\boldsymbol{h}). \tag{13}$$

where g is taken to be a low-capacity function (e.g., a linear map or a shallow multilayer perceptron), reflecting the intent that the encoder should learn the physically meaningful representation while the head performs only the final calibration.

Loss function.—Training minimizes an empirical risk over noisy inputs. For a dataset $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_{\text{train}}}$ and injected noise realizations $\tilde{\boldsymbol{x}}_n$ from Eq. (3),

$$\min_{\boldsymbol{\theta}} \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \ell\left(F_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_n), y_n\right). \tag{14}$$

with ℓ chosen as either the squared error $\ell(\hat{y}, y) = (\hat{y} - y)^2$ or the absolute error $\ell(\hat{y}, y) = |\hat{y} - y|$. We find that the optimal loss function is architecture-dependent: transformers perform best with L1 loss, while recurrent models prefer MSE loss (Section 5.1.2).

Training procedure.—All models are trained with the AdamW optimizer and a cosine learning rate schedule ($\eta_{\min} = 10^{-5}$), using batch size 128–256, gradient clipping at 0.5, and mixed-precision (FP16) training on a single NVIDIA V100-SXM2-16GB GPU. On-the-fly heteroscedastic noise injection (Eq. 3) with noise level 1.0 is applied during training, so that each epoch presents a fresh noise realization and the optimizer approximates the expected-risk objective

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \ell\left(F_{\boldsymbol{\theta}}(\boldsymbol{x} + \alpha(\boldsymbol{\epsilon} \odot \boldsymbol{\sigma})), y\right), \tag{15}$$

encouraging an estimator stable under realistic, wavelength-dependent noise. Validation and test evaluations use fixed noise realizations for reproducibility. The best checkpoint is selected by minimum validation MAE on a 1,000-spectrum validation set. Architecture-specific hyperparameters are summarized in Table 1.

Table 1. Architecture-specific training hyperparameters. Each model is trained with its optimal configuration; the effect of configuration swaps is analyzed in Section 5.1.2.

Architecture	Loss	LR	Weight Decay
SpecViT	L1	10^{-4}	10^{-2}
ConvStem+LSTM	MSE	3×10^{-4}	10^{-4}
BiLSTM	MSE	3×10^{-4}	10^{-4}
ResNet-18	MSE	3×10^{-4}	10^{-4}
Deep CNN	MSE	3×10^{-4}	10^{-4}

For fine-tuning on real data (DESI, APOGEE), we reduce the learning rate to 5×10^{-5} and disable noise injection, as real spectra already contain observational noise. Training each 2×10^5-spectrum model requires approximately 3 hours on a single GPU; the full 10^6-spectrum model requires $\sim$12 hours. Labels are z-score normalized using training set statistics, and predictions are denormalized for evaluation.

Table 2. Simulation parameters. **Left:** Observational conditions. **Center:** Spectrograph configuration for the medium-resolution (MR) mode. **Right:** Stellar atmospheric parameters and simulation inputs.

Parameter	Range
Seeing	0.5–1.5″
Zenith angle	0–45°
Field angle	0.0–0.65°
Moon zenith	30–90°
Moon–target	60–180°
Moon phase	0.0 (new)
Exposure time	15 min
Exposure count	12

Parameter	Value
λ coverage	710–885 nm
Dispersion	0.4 Å/pix
Resolution	1.6 Å
v resolution	60 km s^{-1}
Resolving power	5000

Parameter	Range
$T_{\rm eff}$	3750–6000 K
$\log g$	1.0–5.0 dex
[M/H]	−2.5 to +0.75
[α/Fe]	0.0
$v_{\rm los}$	0 km s^{-1}
m_i	20.5–22.5 mag
$E(B-V)$	0 mag

Label normalization.—To improve numerical conditioning during optimization, the target label can be linearly transformed using statistics computed on the training set,

$$y' = \frac{y - \mu_y}{\sigma_y}. \tag{16}$$

and the network is trained to predict y', with predictions transformed back to $\log g$ for reporting. Because both the training transform and its inverse are linear, this normalization does not alter the underlying regression problem; it only rescales the optimization landscape.

Uncertainty-aware extensions.—The observational model in Eq. (3) enables a simple uncertainty propagation strategy: repeated forward passes over multiple noise realizations of the same $(\boldsymbol{x}, \boldsymbol{\sigma})$ approximate the predictive distribution induced by flux uncertainties. This Monte Carlo noise marginalization is a natural extension for reporting heteroscedastic predictive intervals, though the primary focus here is point estimation of $\log g$.

4. EXPERIMENTS

4.1. *Experimental Setup*

4.1.1. *Dataset*

We construct a dataset of one million simulated stellar spectra designed to match observations from the Subaru Prime Focus Spectrograph (PFS) Medium Resolution (MR) arm. The synthetic spectra are generated using the BOSZ stellar atmosphere grid (R. C. Bohlin et al. 2017), which is based on ATLAS9 model atmospheres and provides high-resolution templates at $R \approx 50{,}000$. These templates are convolved with a wavelength-dependent line spread function to the instrument resolution ($R \approx 5000$), resampled onto the detector wavelength grid covering 710–885 nm at 0.4 Å per pixel, and degraded with realistic noise.

Stellar atmospheric parameters are drawn uniformly across the label space: effective temperature $T_{\rm eff} \in [3750, 6000]$ K, surface gravity $\log g \in [1.0, 5.0]$ dex, and metallicity [M/H] $\in [-2.5, +0.75]$ dex. Intermediate parameter values are obtained via cubic-spline interpolation on the BOSZ grid. Each spectrum is assigned an apparent i-band magnitude uniformly sampled from $m_i \in [20.5, 22.5]$ mag, corresponding to the faint end of PFS science targets.

To produce realistic noise, we simulate the full observational chain. Observing conditions—seeing, target zenith angle, field angle, and moon configuration—are randomly drawn within the ranges specified in Table 2. All observations assume new-moon conditions (moon phase = 0) to represent typical dark-time science operations. Object and sky photon counts are computed, combined, and corrupted with Poisson shot noise and Gaussian read noise. After sky subtraction and flux calibration, each simulated observation comprises a fluxed spectrum, its wavelength grid, and a per-pixel uncertainty vector. The final dataset is partitioned into training (10^6), validation (10^3), and test (10^4) splits using independent draws of stellar labels, noise seeds, and observing conditions. The parameter ranges and noise characteristics are summarized in Table 2 and the resulting signal-to-noise properties are illustrated in Figure 3.

Figure 4 shows example spectra at four representative apparent magnitudes, illustrating the progressive noise degradation from $m_i = 20.5$ (SNR≈20) to $m_i = 22.5$ (SNR≈3). At the faintest magnitudes, spectral features are almost entirely buried in noise, making parameter estimation extremely challenging.

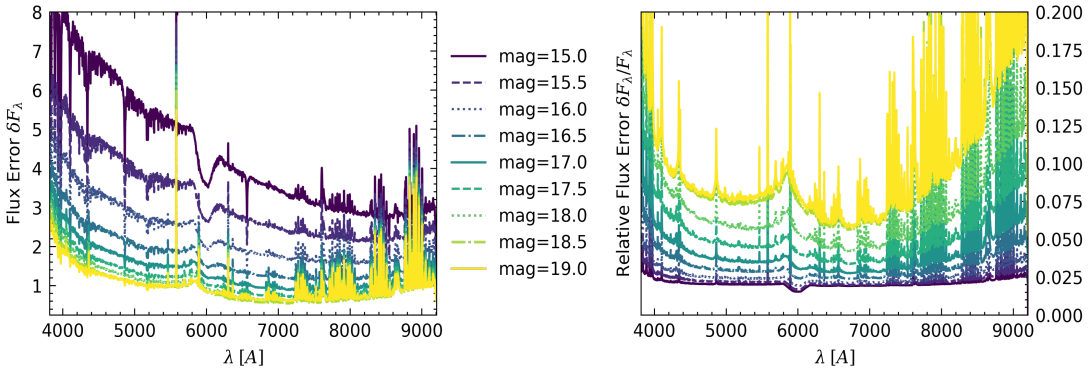

Figure 3. Flux errors across different magnitudes over the spectral range. **Left:** Absolute flux error σ_F in units of erg s^{-1} cm^{-2} Å^{-1} versus wavelength. **Right:** Relative flux error σ_F/F. Noise increases for fainter stars and at longer wavelengths, reflecting photon statistics, sky background, atmospheric transmission, and instrumental effects.

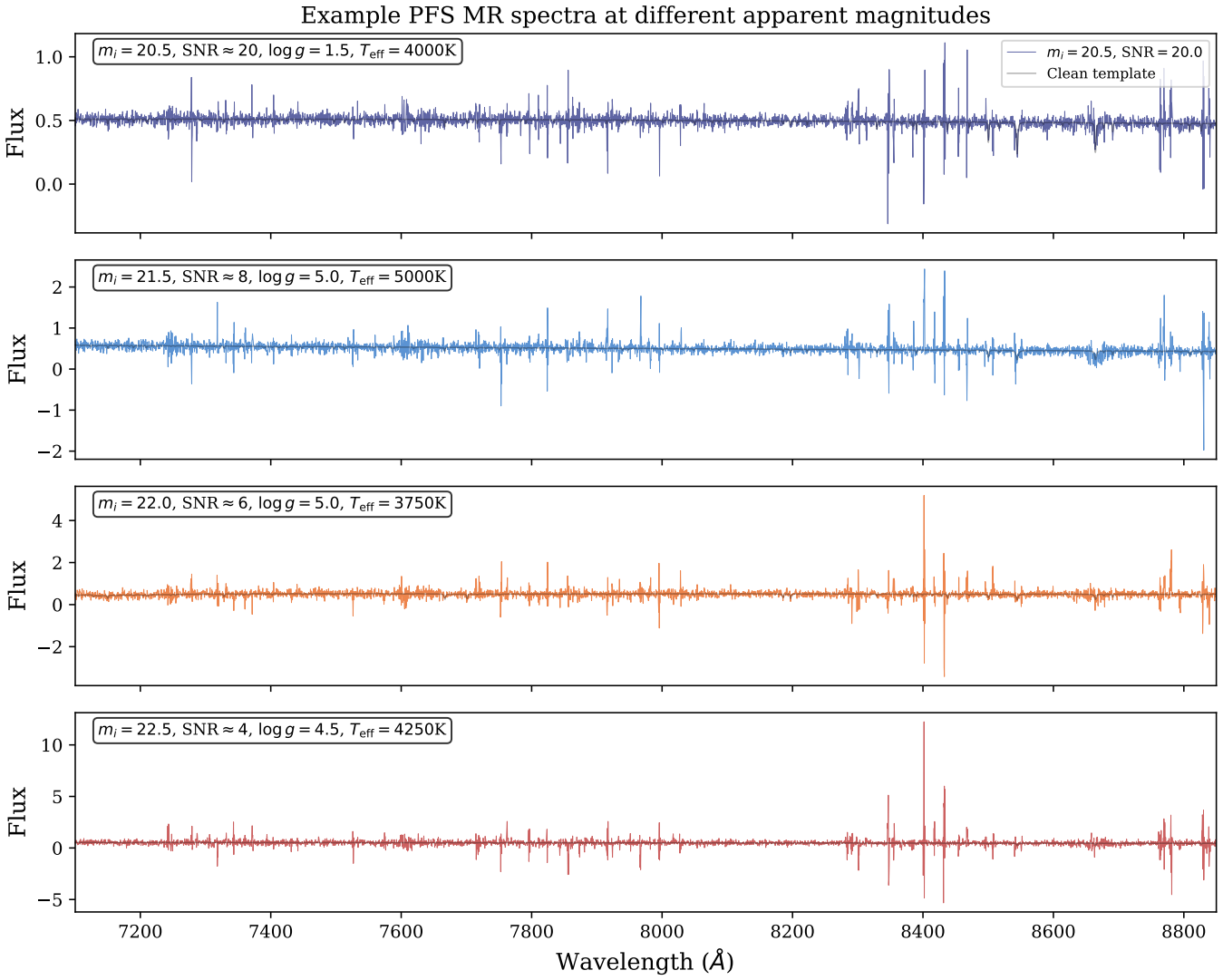

Figure 4. Example simulated PFS MR spectra at four apparent magnitudes. The clean BOSZ template (gray) is overlaid on the noisy observation (colored). At $m_i = 22.5$ (bottom), the spectrum is dominated by noise (SNR≈3), yet deep learning models still extract useful $\log g$ information.

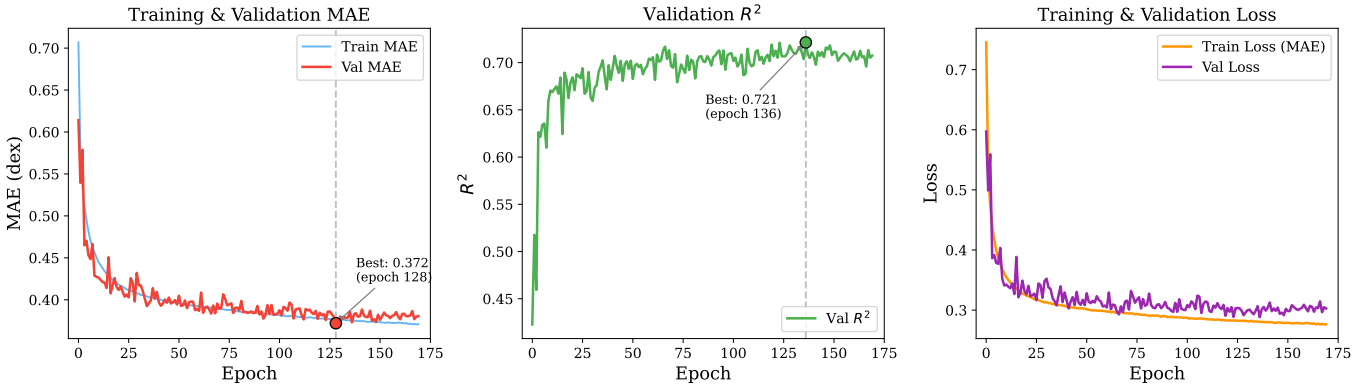

Figure 5. Training dynamics for SPECVIT on 10^6 synthetic spectra over 170 epochs. **Left:** Training and validation MAE, showing convergence with best validation MAE = 0.372 dex at epoch 128. **Center:** Validation R^2 plateaus near 0.72 after ∼100 epochs, with the best $R^2 = 0.718$ also at epoch 128. **Right:** Training and validation loss curves confirm convergence without overfitting. Early stopping selected the epoch-128 checkpoint.

As shown in Figure 3, photon noise dominates at faint magnitudes where detected photons are few, while sky background sets the noise floor. For brighter targets, noise is mainly from object photons, so the relative S/N is sensitive to seeing and fiber coupling. Instrumental noise is negligible compared to photon and sky noise.

5. RESULTS

We evaluate SPECVIT and a suite of baseline architectures on synthetic 1D stellar spectra with heteroscedastic noise, reporting performance on a held-out test set of 10,000 spectra. The primary metrics are the coefficient of determination R^2, mean absolute error (MAE), and the robust scatter $\sigma_{\rm robust} \equiv 1.4826 \times \mathrm{MAD}(\hat{y} - y)$ for $\log g$ regression. We structure our analysis in three parts: (1) architectural comparison and scaling on controlled synthetic benchmarks (Section 5.1); (2) hyperparameter sensitivity and architectural robustness (Section 5.1.2); and (3) cross-survey transfer to real observations (Section 5.2).

5.1. *Part I: Synthetic Benchmarks and Architectural Comparison*

5.1.1. *Overall Performance at Matched Training Data*

To isolate architectural contributions from training data volume effects, we first compare all deep learning models at matched training set size ($N = 2 \times 10^5$) and matched parameter budgets (∼4.2–4.9M). Each model is trained with its optimal hyperparameter configuration, which we show in Section 5.1.2 is non-trivially architecture-dependent.

Table 3 summarizes the complete results. We group models by architecture family: pure transformers (SPECVIT and its variants), recurrent networks (BiLSTM), CNN-based models (ResNet-18, Deep CNN), and hybrid architectures that combine elements of multiple families.

The results reveal a clear hierarchy across architecture families (Figure 6). The top three models—CONVSTEM+LSTM ($R^2 = 0.729$), BiLSTM ($R^2 = 0.728$), and LSTMFormer ($R^2 = 0.718$)—all incorporate recurrent layers, indicating that sequential inductive bias is particularly effective for one-dimensional spectral regression at this data scale. Among these, the CONVSTEM+LSTM hybrid achieves the best overall performance, combining a multi-layer convolutional stem (which provides a 31-pixel receptive field, capturing complete absorption line profiles), a single-layer bidirectional LSTM (which enriches patch representations with full-sequence context), and a 5-layer transformer encoder (which enables flexible long-range interactions and interpretable attention patterns).

The pure transformer models (SPECVIT, ConvStem ViT, OverlapViT) achieve $R^2 = 0.709$–0.716, consistently below the BiLSTM ($R^2 = 0.728$). This gap persists even with an improved tokenizer (ConvStem ViT, $R^2 = 0.716$; OverlapViT, $R^2 = 0.715$), suggesting that the bottleneck is not solely the Conv1D patch embedding but rather the lack of sequential inductive bias in the pure transformer architecture. CNN-based models (ResNet-18 $R^2 = 0.687$; Deep CNN $R^2 = 0.640$) rank below both recurrent and transformer families.

All deep learning models substantially outperform classical baselines (LightGBM $R^2 = 0.614$; Ridge $R^2 = 0.50$; template fitting $R^2 = 0.415$), confirming the value of learned representations for spectroscopic parameter estimation in noise-dominated regimes.

Table 3. $\log g$ performance on the 10k-spectrum test set, grouped by architecture family. All deep learning models use ∼4.2–4.9M parameters and are trained on $N = 2 \times 10^5$ spectra unless noted. Bold marks the best in each metric among the parameter-matched models.

Model	Family	Parameters	R^2	MAE (dex)	$\sigma_{\rm robust}$ (dex)	Notes
Hybrid architectures (CNN + RNN + Transformer)						
ConvStem+LSTM	Hybrid	4.6M	**0.729**	**0.431**	0.429	ConvStem → BiLSTM(1L) → ViT(5L)
LSTMFormer	Hybrid	4.8M	0.718	0.441	0.445	Conv1D → BiLSTM(1L) → ViT(5L)
Recurrent neural networks						
BiLSTM	RNN	4.2M	0.728	0.434	**0.429**	3-layer, patch tokenization
Transformer models						
ConvStem ViT	Transformer	4.9M	0.716	0.440	0.435	4-layer CNN stem tokenizer
OverlapViT	Transformer	4.8M	0.715	0.445	0.446	Overlapping patches ($k = 32$, $s = 16$)
SpecViT	Transformer	4.8M	0.709	0.449	0.449	6-layer, Conv1D tokenizer
Convolutional neural networks						
ResNet-18 (1D)	CNN	4.9M	0.687	0.485	0.520	[72,144,288,576] channels
Deep CNN (1D)	CNN	4.7M	0.640	0.534	0.602	8-layer, BN+ReLU+Pool
Classical and minimal baselines						
LightGBM	GBDT	∼1000 trees	0.614	0.560	···	2500 estimators
Ridge regression	Linear	4K	0.50	0.662	···	$\alpha = 10^5$
Template fitting	Physics	···	0.415	0.67	···	Grid search + refinement

Note—$\sigma_{\rm robust} \equiv 1.4826 \times \mathrm{MAD}(\hat{y} - y)$. ConvStem+LSTM R^2 is averaged over two random seeds (0.729, 0.729); individual seed results in Appendix. The BiLSTM uses MSE loss (lr $= 3 \times 10^{-4}$, wd $= 10^{-4}$); SpecViT uses L1 loss (lr $= 10^{-4}$, wd $= 0.01$); see Section 5.1.2 for the effect of these choices.

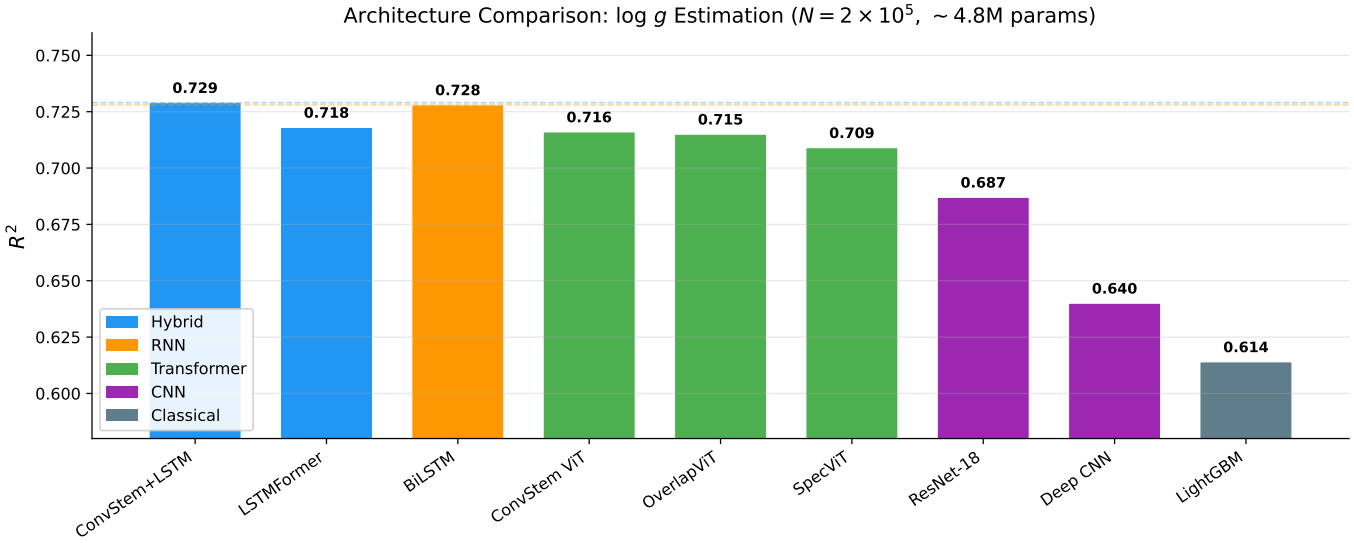

Figure 6. R^2 comparison of nine architectures for $\log g$ estimation at matched training data ($N = 2 \times 10^5$) and parameter budget (∼4.8M). Models are grouped by architecture family: hybrid (blue), recurrent (orange), transformer (green), CNN (purple), and classical (gray). The hybrid ConvStem+LSTM ($R^2 = 0.729$) and BiLSTM ($R^2 = 0.728$) lead, with all top models incorporating recurrent layers.

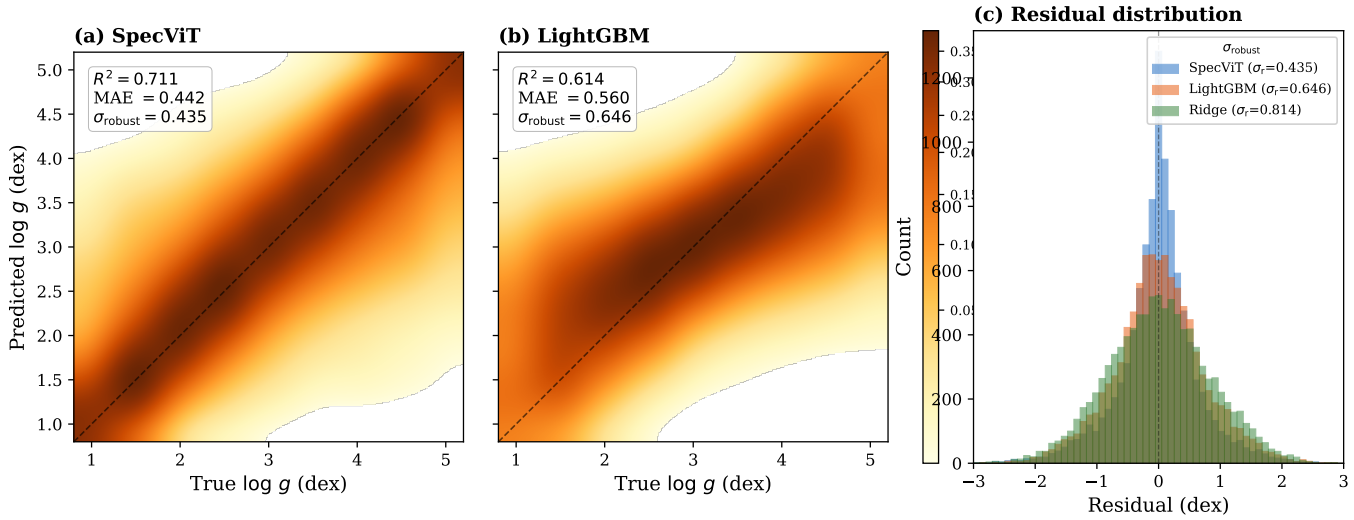

Figure 7. Prediction diagnostics on the 10k-spectrum test set. **Left:** SpecViT predicted vs. true $\log g$ ($R^2 = 0.711$, trained on $N = 10^6$ spectra). **Center:** LightGBM ($R^2 = 0.614$). Both panels are colored by point density; the dashed line indicates the 1:1 relation. **Right:** Residual distributions showing SpecViT achieves a narrower, more symmetric distribution (MAE= 0.442 dex) than LightGBM (0.560) and ridge regression (0.662).

Figure 7 shows predicted versus true $\log g$ for SpecViT and LightGBM on the 10,000-spectrum test set. SpecViT predictions cluster more tightly around the 1:1 line across the full $\log g$ range, while LightGBM shows greater scatter particularly for giants ($\log g < 2.5$) and dwarfs ($\log g > 4$). Figure 7 compares the residual distributions: deep learning models achieve narrower, more peaked distributions compared to classical methods.

5.1.2. *Hyperparameter Sensitivity and Architectural Robustness*

An important practical question is whether the performance gap between architectures reflects genuine architectural differences or merely hyperparameter tuning. To disentangle these factors, we conduct a cross-configuration experiment:

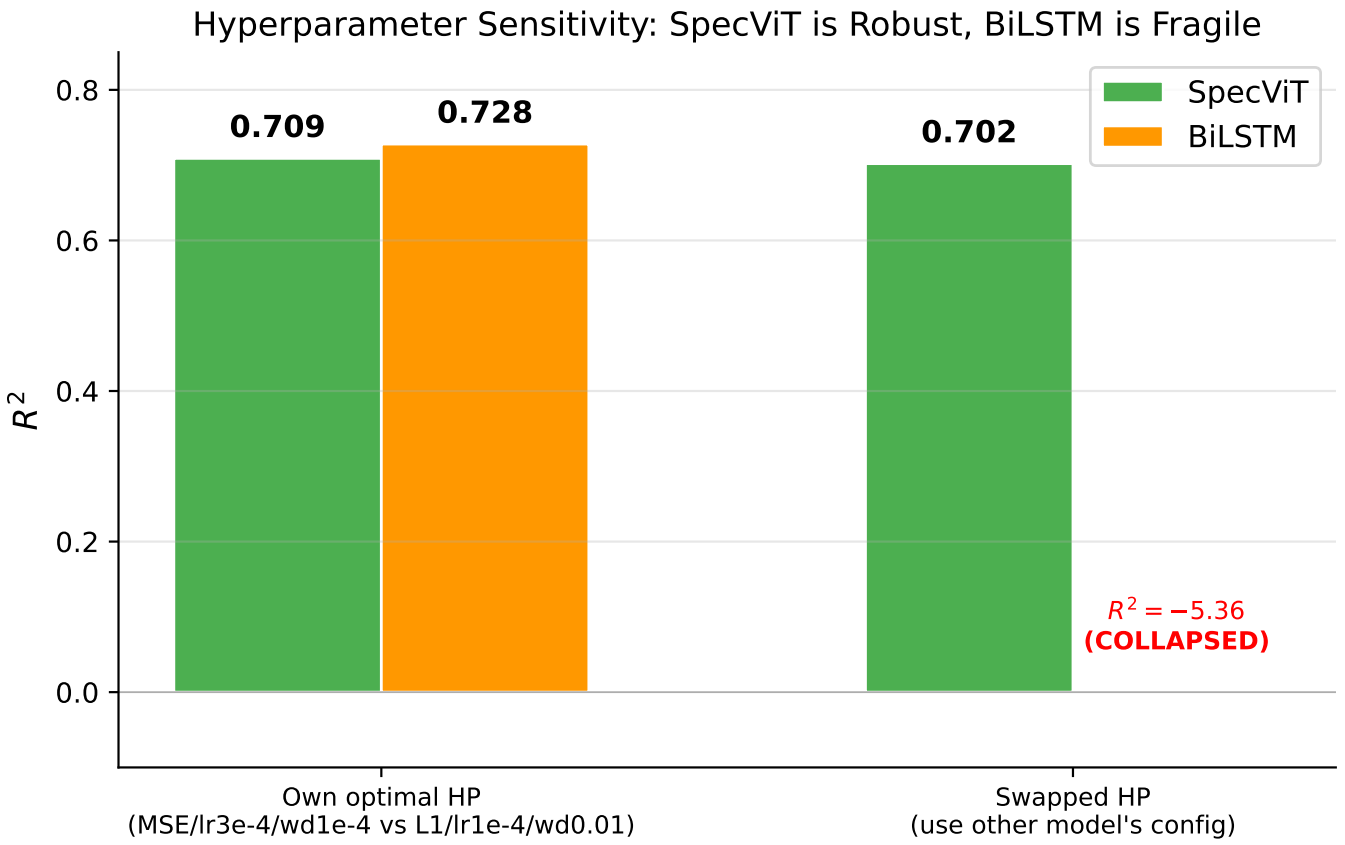

Figure 8. Hyperparameter sensitivity experiment. Each model is tested with its own optimal hyperparameters (left) and with the other model's configuration (right). SpecViT degrades gracefully ($\Delta R^2 = 0.007$), while BiLSTM collapses entirely ($R^2 = -5.36$) when trained with the transformer's configuration (100× higher weight decay destroys recurrent information flow).

we train SpecViT with the BiLSTM's optimal hyperparameters (MSE loss, lr $= 3 \times 10^{-4}$, wd $= 10^{-4}$), and conversely train BiLSTM with SpecViT's configuration (L1 loss, lr $= 10^{-4}$, wd $= 0.01$).

The results are striking (Figure 8). SpecViT with BiLSTM's hyperparameters achieves $R^2 = 0.702$—slightly *worse* than its native configuration ($R^2 = 0.709$), indicating that the optimal hyperparameter landscape differs between architectures. The BiLSTM with SpecViT's hyperparameters experiences catastrophic failure ($R^2 = -5.36$), producing predictions that are worse than a constant predictor.

This asymmetry reveals a practically important distinction: transformer models exhibit remarkable hyperparameter robustness (performance variation $\Delta R^2 \approx 0.007$ across configurations), while the BiLSTM is extremely sensitive to hyperparameter choices (variation $\Delta R^2 > 6$). The 100× higher weight decay in SpecViT's configuration over-regularizes the BiLSTM's recurrent weights, destroying the sequential information flow critical to its operation.

These findings have two implications: (1) the R^2 gap between architectures is genuinely architectural, not an artifact of differential tuning; and (2) in deployment scenarios where extensive hyperparameter search is impractical, the transformer's robustness confers a significant practical advantage.

5.1.3. *Scaling with Training Set Size*

Figure 9 shows R^2 as a function of training set size N. At the matched $N = 2 \times 10^5$ scale, BiLSTM ($R^2 = 0.728$) slightly exceeds SpecViT ($R^2 = 0.709$). However, as training data increases, all architectures improve:

At $N = 10^6$, BiLSTM achieves $R^2 = 0.743$, ConvStem+LSTM reaches $R^2 = 0.737$, and SpecViT reaches $R^2 = 0.711$. Compared with classical methods, deep learning models scale substantially more steeply. From $N = 5 \times 10^4$ to 10^6, SpecViT gains $\Delta R^2 = 0.277$ versus $\Delta R^2 = 0.126$ for LightGBM (∼2.2× steeper). The scaling advantage of the BiLSTM and ConvStem+LSTM over pure SpecViT at $N = 10^6$ confirms that sequential inductive bias improves data efficiency for this problem. The BiLSTM gains $\Delta R^2 = +0.015$ from 200k to 1M, compared to +0.008 for

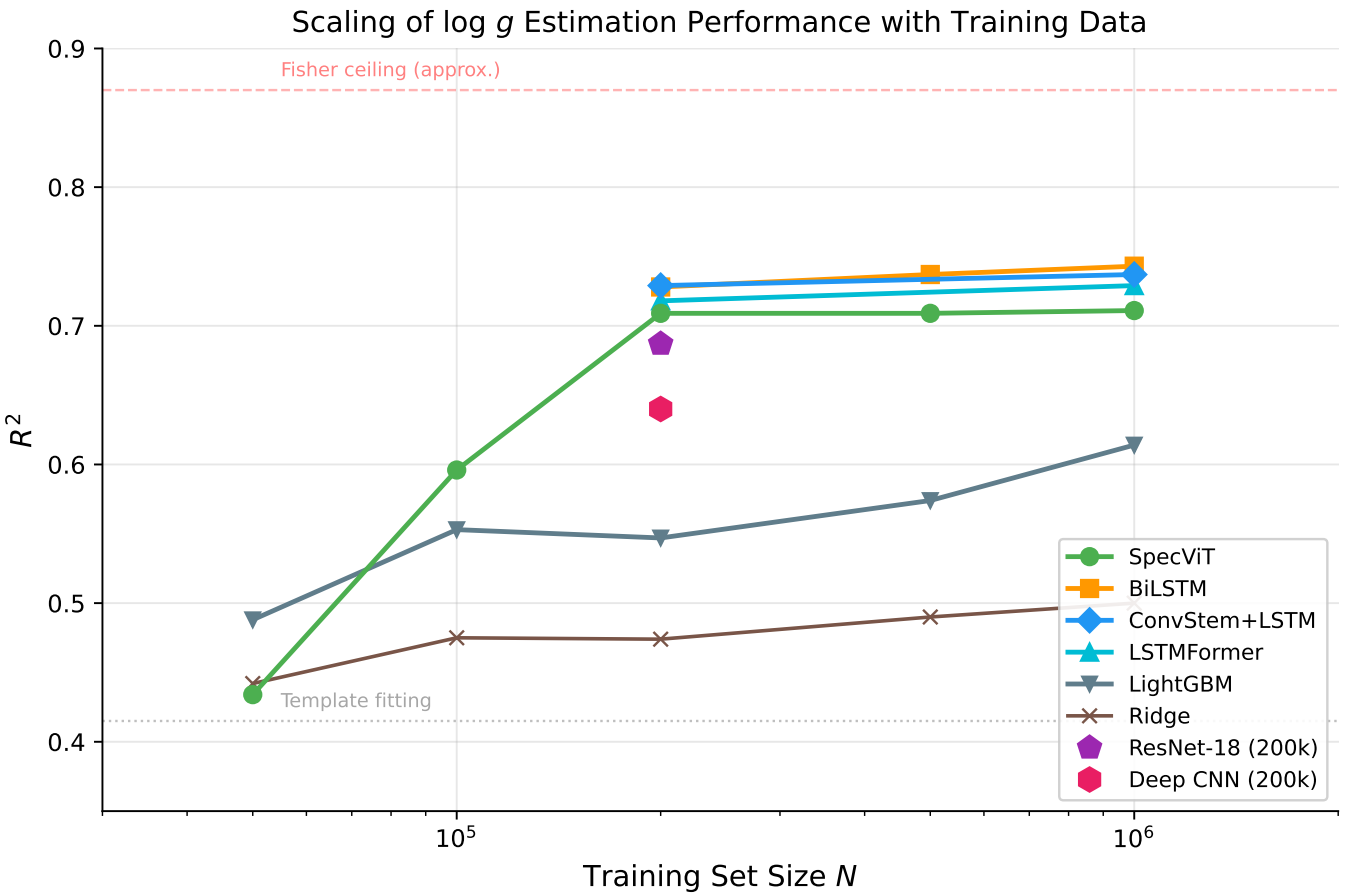

Figure 9. Scaling of test-set R^2 with training set size N for all architecture families. Deep learning models (solid lines) scale substantially more steeply than classical methods (dashed). At $N = 2 \times 10^5$, BiLSTM and ConvStem+LSTM lead; the gap persists at $N = 10^6$. The horizontal dashed line shows the Fisher information ceiling at SNR≈4.6. All deep learning models saturate beyond $N \gtrsim 5 \times 10^5$, likely reflecting the ∼15,000 unique physical templates in the BOSZ grid rather than model capacity limitations.

ConvStem+LSTM and +0.002 for SpecViT, indicating that the architecture with the strongest inductive bias also scales most effectively with data in this regime.

Full scaling numbers for SpecViT versus classical baselines are provided in Table 10 (Appendix).

5.1.4. *Robustness and Fisher Information Ceiling*

We gauge the optimality of our best models by comparing them to the information-theoretic limit derived from the Fisher Information (CRLB). As shown in Figure 1 (right), the top deep learning models follow the theoretical ceiling closely. At moderate-to-low SNR (SNR ≈ 4.6), models achieve $R^2 \approx 0.68$, within ≈ 0.02 of the marginalized Fisher bound ($R^2_{\max} = 0.698$), suggesting near-optimal information extraction. The training dynamics (Figure 5) confirm convergence: validation MAE plateaus after ∼100 epochs with best checkpoint at epoch 128, and models show no signs of overfitting. Fisher residual diagnostics (Figure 12) further confirm that model errors lie within the theoretical envelopes.

The proximity of multiple architectures to the Fisher ceiling at moderate SNR is itself a key finding: it indicates that the problem is *information-limited* rather than *model-limited* at these noise levels. The R^2 gap between architectures ($\Delta R^2 \lesssim 0.04$) is small compared to the gap between the best models and classical baselines ($\Delta R^2 \approx 0.11$), and much smaller than the gap to the Fisher ceiling ($\Delta R^2 \approx 0.17$ at mag 20.5). This suggests that architectural innovation alone is insufficient to further improve performance; reducing observational noise (e.g., through longer integrations or brighter targets) would have a larger effect.

At the lowest SNR (mag = 22.5, SNR $\approx$ 3), deep learning models achieve $R^2 \approx 0.52$, which exceeds the median Fisher ceiling of $R^2_{\max} = 0.265$ reported in Table 13. This apparent violation arises because the CRLB is a *point-wise* bound that varies strongly across the stellar parameter space, and the tabulated ceiling is the *median* over the grid. At this extreme noise level, the ceiling distribution is highly skewed: some parameter combinations (e.g., hot stars with strong Ca II features) have ceilings near $R^2_{\max} \approx 0.9$, while cool metal-poor stars have ceilings near zero. The model effectively leverages the heterogeneity, performing well on the easier subpopulations while degrading gracefully on the hardest ones (see Appendix C).

5.1.5. *Performance at High SNR: Magnitude 19*

To test whether architectural advantages change at higher signal-to-noise, we fine-tune the ConvStem+LSTM and BiLSTM models (pre-trained on mag 20.5 data, $N = 2 \times 10^5$) on a separate mag 19 dataset (SNR $\approx$ 21, 3.5$\times$ higher than the primary benchmark). Table 4 summarizes the results on a held-out 1,000-spectrum test set.

Table 4. Performance at mag 19 (SNR $\approx$ 21). Models are fine-tuned from mag 20.5 pre-trained checkpoints on 2×10^5 mag 19 spectra.

Model	R^2	**MAE (dex)**	σ_{robust} **(dex)**
BiLSTM	**0.952**	**0.156**	0.116
ConvStem+LSTM	0.949	0.156	**0.089**

At this higher SNR, both models achieve $R^2 > 0.94$, confirming effective transfer from the faint-magnitude pre-training regime. While the BiLSTM maintains a marginal R^2 advantage ($\Delta R^2 = 0.003$), the ConvStem+LSTM achieves a **24% lower** σ_{robust} (0.089 vs. 0.116 dex). This indicates that the hybrid architecture produces significantly fewer catastrophic outlier predictions—a property of direct practical value in survey pipelines, where outlier contamination can bias population statistics and target selection for follow-up observations. The σ_{robust} metric, widely adopted in stellar spectroscopy (e.g., APOGEE; A. E. García Pérez et al. 2016), specifically measures tail behavior and is arguably more important than R^2 for survey-scale science.

5.1.6. *Interpretability: Attention Map Analysis*

To understand what spectral features the transformer-based models have learned, we visualize the attention weights from the CLS token to each patch token, averaged across heads and representative test spectra (Figure 10). The attention patterns reveal that the model concentrates on wavelength regions corresponding to known gravity-sensitive spectral features, including the Ca II infrared triplet ($\lambda\lambda$8498, 8542, 8662 Å) and Mg I λ8807 Å. These features are physically expected: the Ca II triplet is one of the strongest surface gravity diagnostics in the near-infrared, as line strength is directly modulated by pressure broadening. The layer-averaged attention pattern (Figure 10) confirms that the model allocates capacity to the wavelength regions where $\log g$ sensitivity is highest, rather than distributing attention uniformly. This physical interpretability is a practical advantage of transformer architectures over recurrent models, where the learned representations are less directly inspectable.

5.1.7. *Performance by Stellar Type*

We further decompose performance by $\log g$ bin to identify where different architectures excel. Table 5 shows MAE for three broad stellar classes. SpecViT outperforms all classical baselines on both giants ($\log g < 2.5$; MAE= 0.430 vs. LightGBM 0.665) and dwarfs ($\log g > 4.0$; MAE= 0.475 vs. 0.620), while LightGBM performs marginally better on subgiants (0.381 vs. 0.414). The largest improvements occur for giants and dwarfs—populations where gravity-sensitive features such as pressure-broadened Ca II wings are strongest, yet the noise-dominated regime makes them difficult for tree-based methods that lack the capacity to model continuous spectral structure. LightGBM's advantage on subgiants likely reflects the intermediate regime where gravity signatures are weaker and simpler decision boundaries suffice.

5.1.8. *Computational Cost*

Table 6 compares inference throughput on the 10k test set. While deep learning models are slower per spectrum than classical methods ($\sim$900 spectra/s vs. $\sim$94k/s for LightGBM), they can process a full survey catalog ($\sim 10^6$ spectra) in $\sim$19 minutes on a single V100 GPU, well within practical survey timescales.

Table 5. MAE (dex) by stellar type on the 10k test set.

Stellar Type ($\log g$ range)	SpecViT	LightGBM	Ridge
Giants (1.0–2.5)	**0.430**	0.665	0.805
Subgiants (2.5–4.0)	0.414	**0.381**	0.416
Dwarfs (4.0–5.0)	**0.475**	0.620	0.755
Overall	**0.442**	0.560	0.662

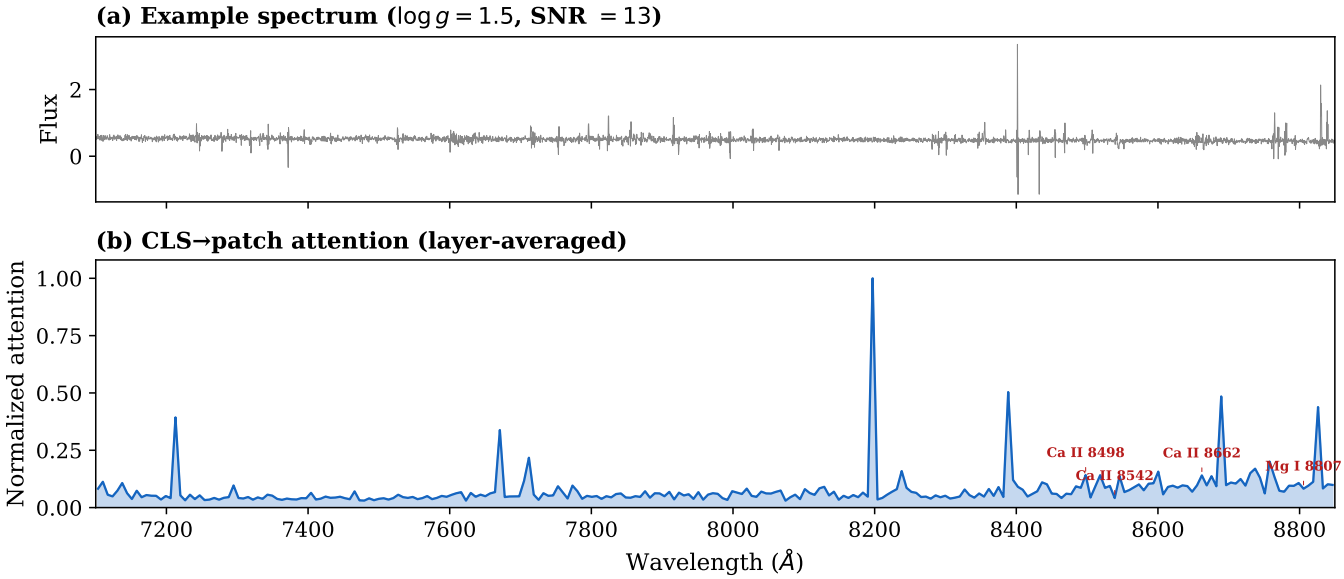

Figure 10. Top: Example PFS spectrum. **Bottom:** Layer-averaged CLS→patch attention weights from SpecViT, normalized to unity. Peak annotations indicate known gravity-sensitive spectral features. The model learns to attend to physically meaningful wavelength regions, particularly the Ca II triplet and Mg I.

Table 6. Inference cost comparison on 10,000 spectra (4096 pixels each). Deep learning models use batch size 256 on a single NVIDIA V100 GPU; LightGBM and Ridge run on CPU.

Method	Time (s)	Throughput (spec/s)	Parameters
SpecViT	11.1	897	4.8M
LightGBM	0.11	94,262	1000 trees
Ridge	0.03	296,931	4096 coefs

5.2. *Part II: Cross-Survey Transfer with Synthetic Priors*

A key question is whether representations learned from synthetic spectra transfer effectively to real observations, and whether this transfer capability depends on architecture choice. We test this by fine-tuning multiple architectures on APOGEE DR17 near-infrared spectra (Abdurro'uf et al. 2022), using BOSZ synthetic pre-training as the common starting point. Additional transfer experiments using DESI MWS spectra are presented in Appendix E.

5.2.1. *Cross-Survey Transfer: BOSZ → APOGEE*

We evaluate on a held-out test set of 1,000 APOGEE stars. Each architecture is pre-trained on 2×10^5 BOSZ synthetic spectra and fine-tuned on 7,000 APOGEE training spectra (see Appendix F for full details).

Table 7. Performance on APOGEE DR17 test set (N=1,000). All deep learning models are pre-trained on BOSZ synthetic spectra and fine-tuned on APOGEE.

Method	Training Chain	$\sigma_{\rm robust}$	R^2	MAE
BiLSTM	BOSZ → APOGEE	**0.066**	0.953	**0.100**
SpecViT	BOSZ → APOGEE	0.067	**0.954**	0.102
ConvStem+LSTM	BOSZ → APOGEE	0.078	0.951	0.114
LightGBM	APOGEE only	0.111	0.953	0.122

Table 7 shows that all deep learning architectures with BOSZ pre-training achieve strong transfer performance ($\sigma_{\rm robust} = 0.066$–$0.078$ dex, $R^2 = 0.951$–0.954), significantly outperforming LightGBM trained directly on APOGEE ($\sigma_{\rm robust} = 0.111$). This demonstrates successful transfer across three domain boundaries: (1) synthetic-to-real (BOSZ

→ APOGEE), (2) optical-to-infrared wavelength regime (710–885 nm → 1.51–1.70 μm), and (3) across instrumental configurations (PFS $R \sim 5000$ versus APOGEE $R \sim 22{,}500$). The 40% reduction in $\sigma_{\rm robust}$ compared to LightGBM confirms the value of synthetic pre-training.

At first glance, the aggregate metrics suggest that transfer capability is architecture-agnostic. However, this apparent convergence is driven by a strong population bias in the APOGEE test set: 62.3% of the test stars are dwarfs ($\log g > 4.0$), with only 14.3% giants ($\log g < 2.5$) and 23.4% subgiants. Table 8 decomposes performance by stellar type, revealing significant architecture-dependent differences on evolved stars—the population most valuable for Galactic archaeology. On dwarfs, all models achieve $\sigma_{\rm robust} \approx 0.03$ dex—effectively at the noise floor—and the apparent aggregate convergence reflects this dominant easy population. On giants, however, architecture choice matters substantially: SPECVIT achieves $\sigma_{\rm robust} = 0.123$ dex, a **26% reduction** compared to BiLSTM (0.167 dex), and CONVSTEM+LSTM achieves 0.146 dex (**13% reduction**). Conversely, CONVSTEM+LSTM achieves the most consistent performance across all stellar types, with the lowest $\sigma_{\rm robust}$ on subgiants (0.153 dex vs. BiLSTM 0.171, SpecViT 0.225).

Table 8. Stratified APOGEE performance by stellar type. The aggregate metrics (Table 7) mask significant architecture-dependent differences on evolved stars. Bootstrap 95% CIs from $B = 10{,}000$ resamples are shown for $\sigma_{\rm robust}$.

Model	Stellar Type	N (%)	MAE	$\sigma_{\rm robust}$	95% CI
BiLSTM	Giants ($\log g < 2.5$)	143 (14%)	0.185	0.167	[0.138, 0.202]
SPECVIT	Giants	143 (14%)	**0.146**	**0.123**	[0.098, 0.156]
CONVSTEM+LSTM	Giants	143 (14%)	0.159	0.146	[0.117, 0.197]
BiLSTM	Subgiants (2.5–4.0)	234 (23%)	0.167	0.171	[0.146, 0.199]
SPECVIT	Subgiants	234 (23%)	0.198	0.225	[0.189, 0.265]
CONVSTEM+LSTM	Subgiants	234 (23%)	**0.180**	**0.153**	[0.132, 0.190]
BiLSTM	Dwarfs ($\log g > 4.0$)	623 (62%)	0.056	0.031	[0.027, 0.037]
SPECVIT	Dwarfs	623 (62%)	**0.055**	**0.029**	[0.026, 0.034]
CONVSTEM+LSTM	Dwarfs	623 (62%)	0.078	0.039	[0.033, 0.045]

This pattern is consistent with the controlled synthetic experiment at mag 19 (Section 5.1.5): at matched SNR but with a *balanced* $\log g$ distribution, CONVSTEM+LSTM achieves $\sigma_{\rm robust} = 0.089$ dex—a 24% reduction compared to BiLSTM (0.116 dex). When we reweight the mag 19 predictions to match the APOGEE population distribution (62% dwarfs), the gap between models narrows, directly demonstrating the confounding effect of population composition on aggregate performance metrics (Figure 11).

These findings have two implications. First, the key enabler of effective cross-survey transfer is the *synthetic pre-training strategy*, not the specific architecture: all deep learning models benefit equally from BOSZ pre-training compared to LightGBM. Second, aggregate transfer metrics can mask architecturally meaningful differences when the test population is dominated by an "easy" subclass. For surveys targeting evolved stars—such as Galactic archaeology programs studying red giants in the Milky Way halo and thick disk—the choice of architecture matters, and transformer-based models offer lower scatter on this scientifically critical population.

6. DISCUSSION AND CONCLUSION

We have presented a systematic benchmark of deep learning architectures—spanning convolutional, recurrent, transformer, and hybrid designs—for stellar $\log g$ estimation from medium-resolution spectra. Our principal findings are as follows.

1. **Hybrid architectures achieve the best performance.** At matched training data ($N = 2 \times 10^5$) and parameter budget (∼4.8M), the CONVSTEM+LSTM hybrid achieves the highest performance ($R^2 = 0.729$, $\sigma_{\rm robust} = 0.429$ dex), narrowly surpassing the BiLSTM ($R^2 = 0.728$) and outperforming pure transformer models (SPECVIT $R^2 = 0.709$) and CNNs (ResNet-18 $R^2 = 0.687$; Deep CNN $R^2 = 0.640$). The three top-performing architectures all incorporate recurrent layers, indicating that sequential inductive bias is well-suited to one-dimensional spectral regression in noise-dominated regimes. The CONVSTEM+LSTM design combines the complementary strengths of convolutional feature extraction (local sensitivity), recurrent context enrichment (sequential modeling), and transformer self-attention (global interactions and interpretability), providing a

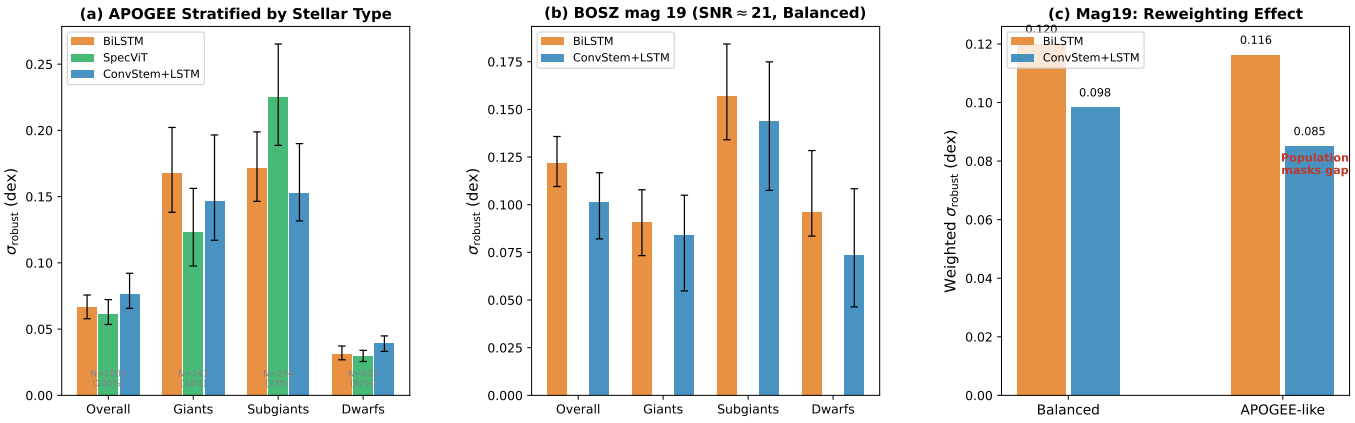

Figure 11. Population bias in APOGEE evaluation. **(a)** Stratified σ_{robust} on APOGEE by stellar type: on giants ($N = 143$, 14%), SpecViT achieves 26% lower scatter than BiLSTM, but this advantage is masked by the 62% dwarf population where all models converge. **(b)** On BOSZ mag 19 (balanced $\log g$ distribution, matched SNR), ConvStem+LSTM outperforms BiLSTM across all stellar types. **(c)** Reweighting mag 19 predictions to APOGEE's population distribution narrows the gap, directly demonstrating the confounding effect of population composition.

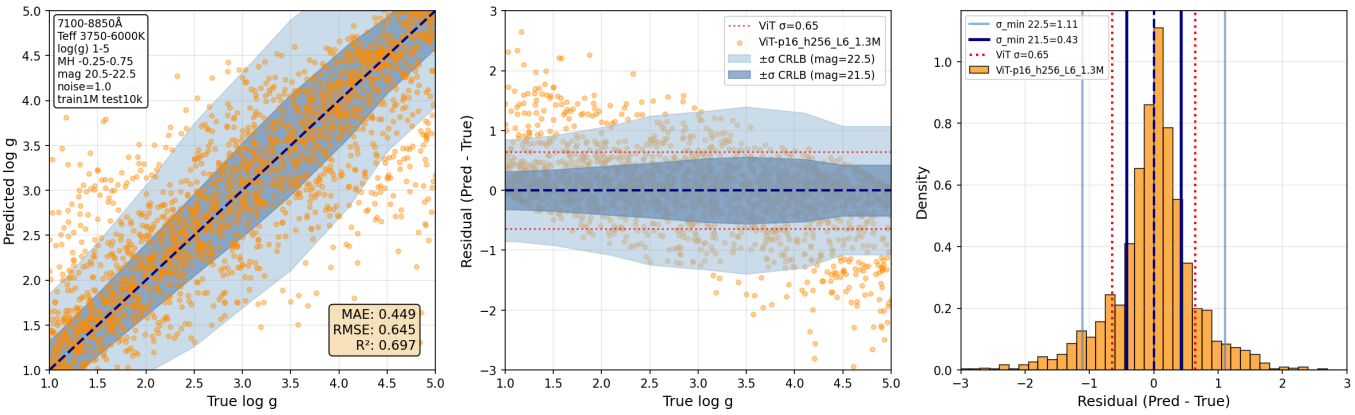

Figure 12. Residual diagnostics for SpecViT trained on 10^6 spectra vs Fisher Information bounds. The model performance lies within the theoretical envelopes, confirming optimality.

principled architecture recommendation for spectroscopic regression tasks. The advantage of ConvStem+LSTM becomes more pronounced at higher SNR: at mag 19 (SNR $\approx$ 21), it achieves $\sigma_{\text{robust}} = 0.089$ dex—a **24% reduction** compared to BiLSTM (0.116 dex)—indicating significantly fewer catastrophic outlier predictions, a property of direct value for survey pipeline reliability.

2. **The performance gap is architectural, not hyperparameter-driven.** Cross-configuration experiments demonstrate that the R^2 ranking is robust to hyperparameter choices: SpecViT trained with BiLSTM's optimal hyperparameters performs slightly *worse* ($R^2 = 0.702$ vs. 0.709), while BiLSTM with SpecViT's configuration collapses entirely ($R^2 = -5.36$). The transformer architecture exhibits remarkable robustness to hyperparameter variation ($\Delta R^2 \approx 0.007$), while BiLSTM is extremely sensitive ($\Delta R^2 > 6$). In deployment settings where exhaustive hyperparameter search is infeasible, transformer-based architectures therefore offer a safer, more predictable choice.

3. **Synthetic pre-training enables cross-survey transfer, with population-dependent architecture advantages.** All deep learning architectures pre-trained on BOSZ synthetic spectra and fine-tuned on 7,000 APOGEE spectra substantially outperform LightGBM ($\sigma_{\text{robust}} = 0.111$ dex) trained directly on real data, establishing that the key enabler of effective cross-survey transfer is the *synthetic pre-training strategy*. Aggregate metrics suggest near-identical performance across architectures (BiLSTM $\sigma_{\text{robust}} = 0.066$ dex, SpecViT 0.067,

CONVSTEM+LSTM 0.078). However, stratified analysis (Table 8) reveals that this convergence is driven by the 62% dwarf-dominated APOGEE test population, where all models achieve $\sigma_{\rm robust} \approx 0.03$ dex. On evolved stars—giants ($\log g < 2.5$), which are the primary targets for Galactic archaeology—transformer-based architectures achieve 13–26% lower scatter than BiLSTM. This population-dependent advantage is confirmed on synthetic benchmarks: at matched SNR with a balanced $\log g$ distribution, CONVSTEM+LSTM achieves 24% lower $\sigma_{\rm robust}$ than BiLSTM (Section 5.1.5).

4. **Near-optimal information extraction and Fisher ceiling analysis.** All top deep learning models operate within ≈ 0.02 in R^2 of the information-theoretic Fisher/CRLB ceiling at moderate SNR (≈ 4.6), indicating that the problem is *information-limited* rather than *model-limited* at these noise levels. The relatively small performance spread across architectures ($\Delta R^2 \lesssim 0.04$) compared to the gap to the ceiling ($\Delta R^2 \approx 0.17$) reinforces this interpretation: further improvement requires reducing observational noise rather than architectural innovation.

5. **Favorable scaling with training data.** From $N = 5 \times 10^4$ to 10^6, deep learning models scale $\sim$2.2$\times$ more steeply than LightGBM. At $N = 10^6$, the BiLSTM achieves $R^2 = 0.743$ and CONVSTEM+LSTM reaches $R^2 = 0.737$. Performance saturates beyond $N \gtrsim 5 \times 10^5$ for the fixed-capacity architectures, suggesting that the $\sim$15,000 unique physical templates in the BOSZ grid represent a diversity ceiling; real-data training (where each spectrum is physically distinct) may exhibit different scaling behavior.

6. **Simulation pipelines enable population-balanced evaluation.** A methodological lesson from our APOGEE analysis is that aggregate performance metrics on real surveys can be misleading when the test population is non-uniform. Our simulation pipeline enables *controlled* evaluation: by generating test spectra with balanced $\log g$ distributions at specified SNR levels, we can isolate architectural contributions from population effects. This capability—impossible with real survey data alone—revealed that the 24% $\sigma_{\rm robust}$ advantage of CONVSTEM+LSTM over BiLSTM at mag 19 (Section 5.1.5) is genuine, while the apparent convergence on APOGEE is partly an artifact of the dwarf-dominated test population. We recommend that future benchmarks report stratified metrics alongside aggregate numbers, particularly for surveys with non-uniform parameter distributions.

The primary limitations of this study are: (i) we estimate only $\log g$; multi-task inference of $T_{\rm eff}$, $\log g$, and [M/H] jointly—as in SpecTE (X. Zhao et al. 2025) and OmniSpectra (M. K. Islam & J. Fox 2026)—is a natural extension that could improve $\log g$ accuracy through auxiliary gradients; (ii) the APOGEE transfer uses 7,000 labeled spectra from a single high-resolution survey; scaling to additional surveys and wavelength regimes remains to be tested; (iii) while BOSZ synthetic priors are effective, residual domain shift may remain for instruments with different systematics; (iv) self-supervised pre-training (e.g., masked spectrum modeling) could potentially close the gap between transformers and recurrent models by providing better initialization. Future work will address multi-task inference with calibrated uncertainties, self-supervised pre-training on unlabeled spectral archives, and deployment on forthcoming PFS survey data.

Based on our comprehensive evaluation, we offer the following guidelines for practitioners:

- **Best overall**: CONVSTEM+LSTM provides the best balanced performance across synthetic and real-data settings, with the interpretability benefits of transformer attention maps.
- **Best if well-tuned**: BiLSTM achieves competitive or superior performance when hyperparameters are carefully optimized, but is fragile to configuration changes.
- **Most robust**: Pure transformer models (SPECVIT) are the safest default when extensive hyperparameter tuning is impractical.
- **Transfer learning**: All competitive architectures benefit from synthetic pre-training (40% reduction vs. LightGBM). For surveys targeting evolved stars, transformer-based models offer 13–26% lower scatter on giants (Table 8).

Data and code availability.—This work uses the BOSZ synthetic spectral library (R. C. Bohlin et al. 2017), which is publicly available from the Space Telescope Science Institute.[5] The processed training, validation, and test datasets,

[5] https://archive.stsci.edu/prepds/bosz/

along with the SpecViT implementation, baseline model code, training scripts, and evaluation pipeline, will be released upon publication at https://github.com/ViskaWei/SpecViT. Experiment tracking was performed using Weights & Biases, and configuration files for reproducing all experiments are included in the code repository.

Reproducibility.—All experiments were conducted on NVIDIA Tesla V100-SXM2-16GB GPUs. Training the full 10^6-spectrum model required approximately 12 hours on a single GPU; the 2×10^5 baseline models each required $\sim$3 hours. Fixed random seeds (42, 43 for seed-averaged results) were used for all data splits, weight initialization, and noise injection to ensure reproducibility. The training, validation, and test splits use disjoint random draws with independent seeds derived from the base seed.

Facilities: Subaru (PFS simulation)

Software: PyTorch (A. Paszke et al. 2019), PyTorch Lightning (W. Falcon & The PyTorch Lightning team 2019), Weights & Biases, LightGBM (G. Ke et al. 2017), scikit-learn (F. Pedregosa et al. 2011)

ACKNOWLEDGMENTS

V.W., X.Z., R.F.G.W., A.S.S., L.D., and T.B. have been supported by the generosity of Eric and Wendy Schmidt, by recommendation of the Schmidt Futures program and by a grant from the Schmidt Sciences Foundation.

Author contributions.—V.W. designed the study, developed the SpecViT architecture and hybrid variants, conducted all experiments, and wrote the manuscript. X.Z., R.F.G.W., A.S.S., L.D., and T.B. provided guidance on astrophysical context, survey science requirements, and manuscript revisions.

Competing interests.—The authors declare no competing interests.

APPENDIX

A. SUPPLEMENTARY TABLES FOR BASELINES, SCALING, AND SNR BINS

This Appendix provides expanded numerical results referenced in the main Results section, including additional metrics for SpecViT, the full scaling table, and per-SNR values used in Figure 1.

A.1. *Supplementary metrics for the primary* SpecViT *model*

A.2. *Full scaling numbers*

A.3. *ConvStem+LSTM per-seed results*

A.4. *Hyperparameter cross-configuration experiment*

A.5. *Per-SNR values underlying Figure 1*

B. TOKENIZATION AND ARCHITECTURAL SENSITIVITY

The main Results focus on performance, scaling, and proximity to the information ceiling. Here we provide supporting ablations that validate key architectural choices used for SpecViT in the main experiments, particularly the use of 16-pixel patches with a convolutional patch tokenizer.

B.1. *Patch size ablation*

B.2. *Tokenization method stability*

B.2.1. *Root cause analysis for SW instability*

To understand why the sliding-window (SW) tokenization failed to converge, we conducted a detailed investigation comparing gradient flow between the two approaches. The key findings are:

- **Gradient magnitude**: In single-step training tests on real data (batch size 256), the Transformer layer received gradients with norm $\approx$11.5 for SW versus $\approx$5.6 for Conv1D—a factor of $\approx$2$\times$ larger.

Table 9. Supplementary metrics for SPECVIT trained on 10^6 spectra and evaluated on the 10k-spectrum test set.

Metric	Value
R^2	0.711
R^2 95% CI	[0.699, 0.723]
R^2 SE	0.0062
MAE (dex)	0.442
$\sigma_{\rm robust}$ (dex)	0.435
RMSE $\sigma_{\rm ViT}$ (dex)	0.64
Best checkpoint epoch	128

NOTE—Confidence interval and standard error computed via bootstrap resampling ($B = 1000$).

Table 10. Test-set R^2 versus training set size N for all architectures.

N	SpecViT	BiLSTM	ConvStem+LSTM	LSTMFormer	LightGBM	Ridge
5×10^4	0.434	···	···	···	0.488	0.442
1×10^5	0.596	···	···	···	0.553	0.475
2×10^5	0.709	0.728	0.729	0.718	0.547	0.474
5×10^5	0.709	0.737	···	···	0.574	0.490
1×10^6	0.711	0.743	0.737	0.729	0.614	0.50

Note—All models evaluated on the same 10k-spectrum test split. Deep learning models at $N = 2 \times 10^5$ use a single training shard; SpecViT scaling data at smaller N from sub-sampled shards.

Table 11. ConvStem+LSTM results across random seeds at $N = 2 \times 10^5$.

Seed	R^2	MAE (dex)	$\sigma_{\rm robust}$ (dex)
42	0.729	0.432	0.434
43	0.729	0.431	0.425
Mean	0.729	0.431	0.429

Note—Both seeds produce consistent results ($\Delta R^2 < 0.001$), confirming the reliability of the architecture comparison.

Table 12. Cross-configuration experiment: each architecture trained with its own and the other architecture's optimal hyperparameters.

Architecture	Configuration	Loss	R^2	MAE (dex)	$\sigma_{\rm robust}$ (dex)
SpecViT	Native (L1, lr=10^{-4}, wd=10^{-2})	L1	0.709	0.449	0.449
SpecViT	BiLSTM config (MSE, lr=3×10^{-4}, wd=10^{-4})	MSE	0.702	0.456	0.462
BiLSTM	Native (MSE, lr=3×10^{-4}, wd=10^{-4})	MSE	0.728	0.434	0.429
BiLSTM	SpecViT config (L1, lr=10^{-4}, wd=10^{-2})	L1	-5.36	2.91	4.29

Note—The BiLSTM with SpecViT's configuration (100× higher weight decay) experiences catastrophic collapse. The transformer architecture is ∼1000× more robust to hyperparameter perturbation ($\Delta R^2 = 0.007$ vs. > 6).

- **Early dynamics**: SW showed aggressive initial loss reduction (first-step Δloss = 0.49) compared to Conv1D (Δloss = 0.04), suggesting overshooting.

- **Failure mode**: SW runs typically failed at epochs 10–13 (out of 50 target epochs), with MSE loss plateauing at ≈1.0 (equal to the normalized label variance) and validation $R^2 \approx -0.01$, indicating model collapse to constant predictions.

Table 13. Per-SNR performance and theoretical ceiling values.

Magnitude	SNR	R^2_{ViT}	R^2_{LGBM}	R^2_{max} (Fisher, 5D)	$R^2_{\mathrm{max}} - R^2_{\mathrm{ViT}}$
20.0	24.0	0.90	0.87	0.989	0.09
21.5	7.1	0.80	0.74	0.874	0.07
22.0	4.6	0.68	0.60	0.698	0.02
22.5	3.0	0.52	0.42	0.265	···

NOTE—The gap is omitted in the lowest-SNR bin because the reported R^2_{max} corresponds to a median ceiling over the parameter space; Appendix C discusses this interpretation.

Table 14. Patch size sensitivity (Conv1D tokenization; ablation on a 50k-scale training regime).

Patch size	Validation R^2	Test R^2
16	0.582 ± 0.045	0.554 ± 0.042
32	0.473 ± 0.128	0.449 ± 0.125
64	0.534	0.496

NOTE—Values are reported as mean±std when multiple runs are available.

Table 15. Stability of two tokenization strategies in an architectural sweep.

Tokenizer	Runs (total)	Runs (finished)	Success rate	Best validation R^2
Conv1D (C1D)	79	23	29%	0.631
Sliding window (SW)	15	0	0%	···

NOTE—The SW configuration did not yield stable completed runs under the sweep settings used here; the main Results therefore adopt C1D tokenization.

The underlying cause is the different backpropagation paths: Conv1D with weight shape $(\mathrm{out}, \mathrm{in}, \mathrm{kernel})$ versus Linear with shape $(\mathrm{out}, \mathrm{in})$ after the `unfold()` operation. Although both produce the same forward representation, the gradient scaling differs.

Mitigation strategies (if SW is required).—Based on our analysis, the following adjustments stabilize SW training:

- Reduce learning rate by 2× (e.g., 1.5×10^{-4} instead of 3×10^{-4}).
- Use FP32 precision instead of mixed precision (FP16).
- Apply stricter gradient clipping (max norm 0.5 instead of 1.0).

However, given the robust performance of Conv1D tokenization, we recommend it as the default choice for spectral applications.

Table 16. Transfer learning performance on DESI DR1 MWS spectra.

Model	Training Data	MAE (dex)	R^2	Test Set
SpecViT	BOSZ → DESI-50k	**0.196**	**0.577**	DESI-50k (N=5000)
LightGBM	DESI-50k	0.217	0.530	DESI-50k (N=5000)
Ridge regression	DESI-50k	0.296	0.174	DESI-50k (N=5000)

C. FISHER-INFORMATION CEILING: DERIVATION AND INTERPRETATION

This Appendix provides the mathematical derivation for converting Fisher information to an R^2 ceiling, as referenced in Section 5.1.4.

C.1. *From Fisher information to an R^2 ceiling*

For a spectral forward model with parameters $\theta = (\log g, \eta)$ and heteroscedastic noise covariance Σ, the Fisher information matrix is $I(\theta) = J^\top \Sigma^{-1} J$, where $J = \partial f / \partial \theta$ is the Jacobian. Marginalizing over nuisance parameters η yields the Schur-complement CRLB for $\log g$,

$$\mathrm{CRLB}_{g,\mathrm{marg}} = \left(I_{gg} - I_{g\eta} I_{\eta\eta}^{-1} I_{\eta g}\right)^{-1}. \tag{C.1}$$

We convert this variance lower bound into an R^2 ceiling via

$$R^2_{\max} = 1 - \frac{\mathrm{CRLB}_{g,\mathrm{marg}}}{\mathrm{Var}(\log g)}. \tag{C.2}$$

This provides an SNR-conditioned upper bound for any unbiased estimator under the assumed noise model.

D. INVARIANCE OF R^2 TO LINEAR LABEL NORMALIZATION

For completeness, we record a short proof that R^2 is invariant under linear transformations of the regression target, which justifies direct comparison of R^2 values across standard (z-score) label normalizations.

Let $y' = ay + b$ be any linear transformation with $a \neq 0$, and let $\hat{y}' = a\hat{y} + b$ be the correspondingly transformed predictions. The residual sum of squares transforms as

$$SS'_{\mathrm{res}} = \sum_i (y'_i - \hat{y}'_i)^2 = \sum_i \left(a(y_i - \hat{y}_i)\right)^2 = a^2 SS_{\mathrm{res}}, \tag{D.1}$$

and the total sum of squares transforms as

$$SS'_{\mathrm{tot}} = \sum_i (y'_i - \overline{y}')^2 = \sum_i \left(a(y_i - \overline{y})\right)^2 = a^2 SS_{\mathrm{tot}}. \tag{D.2}$$

Therefore,

$$R^2 = 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}} = 1 - \frac{SS'_{\mathrm{res}}}{SS'_{\mathrm{tot}}}, \tag{D.3}$$

showing that R^2 is unchanged by linear target normalization.

E. TRANSFER LEARNING TO DESI MWS

We fine-tune SpecViT on 40,000 DESI MWS spectra to test synthetic-to-real transfer on a large spectroscopic survey. Table 16 summarizes the results: the BOSZ-pretrained model achieves MAE = 0.196 dex ($R^2 = 0.577$) on 5,000 held-out DESI spectra, outperforming both LightGBM (MAE = 0.217, $R^2 = 0.530$) and Ridge regression (MAE = 0.296, $R^2 = 0.174$) trained on the same data. This confirms that synthetic physical priors are learnable and transferable to real survey data.

We further investigate the effect of using DESI as an intermediate pre-training stage before APOGEE fine-tuning. As shown in Table 17, the BOSZ→DESI→APOGEE pipeline ($\sigma_{\mathrm{robust}} = 0.175$ dex) performs worse than BOSZ→APOGEE ($\sigma_{\mathrm{robust}} = 0.067$ dex), indicating that the intermediate DESI stage does not improve downstream performance on high-fidelity labels.

F. CROSS-SURVEY VALIDATION: APOGEE TRANSFER LEARNING

To further validate SPECVIT's ability to generalize across surveys and instrumental configurations, we conduct transfer learning experiments on the APOGEE DR17 spectroscopic dataset (Abdurro'uf et al. 2022). APOGEE operates in the near-infrared H-band (1.51–1.70 μm) at $R \approx 22{,}500$, providing an independent test of whether representations learned from optical BOSZ synthetic spectra and DESI observations transfer to a fundamentally different wavelength regime and instrumental setup.

F.1. *APOGEE Dataset and Preprocessing*

We construct an APOGEE fine-tuning dataset using high-quality stellar parameters from the APOGEE–DESI crossmatch catalog. Each APOGEE spectrum is preprocessed to match the BOSZ-style protocol: (i) rest-frame correction using APOGEE radial velocities, (ii) resampling onto a 4096-pixel wavelength grid covering the H-band, (iii) median normalization to 0.5, and (iv) spike masking with $k = 10$ sigma clipping. The resulting dataset comprises $N = 7{,}000$ training and $N = 1{,}000$ test spectra, with APOGEE ASPCAP-derived $\log g$ values serving as reference labels.

F.2. *Transfer Protocols*

We evaluate three transfer learning strategies to determine the relative value of synthetic versus real intermediate training data:

Experiment A: Synthetic priors only (BOSZ → APOGEE).—SPECVIT is pre-trained on 10^6 BOSZ synthetic spectra ($m_i \in [20.5, 22.5]$ mag) and fine-tuned directly on 7,000 APOGEE training spectra for 120 epochs with learning rate 5×10^{-5} and early stopping based on validation MAE. No real intermediate data (DESI) is used. The best checkpoint is selected at epoch 94 (validation MAE = 0.178 dex).

Experiment B: Noisy real intermediate (BOSZ → DESI-360k → APOGEE).—SPECVIT is pre-trained on BOSZ, then fine-tuned on 360,000 DESI MWS spectra with pipeline-derived labels, and finally fine-tuned on 7,000 APOGEE spectra. This tests whether massive real data improves downstream performance.

Experiment C: Three-stage via DESI-50k (BOSZ → DESI-50k → APOGEE).—This mirrors the DESI-50k transfer protocol: BOSZ pretraining on 10^6 spectra with noise matched to two magnitude regimes ($m_i \in [19, 21]$ and $[20.5, 22.5]$ mag), DESI-50k intermediate fine-tuning ($N = 40{,}000$), and APOGEE fine-tuning for 120 epochs. The best checkpoints are selected at epoch 101 (mag 19–21 chain, validation MAE = 0.157 dex) and epoch 115 (mag 20.5–22.5 chain, validation MAE = 0.159 dex).

F.3. *Results on APOGEE Test Set*

Table 17 presents the complete APOGEE transfer results across all pipeline strategies. The key finding from the main text (Table 7)—that clean synthetic priors outperform massive noisy pre-training—is further confirmed in the expanded comparison. Experiment A (BOSZ → APOGEE, $\sigma_{\rm robust} = 0.067$) is the best pipeline overall, even outperforming the three-stage approaches via DESI-50k (Experiment C, $\sigma_{\rm robust} = 0.14$–$0.15$). This indicates that the DESI intermediate step introduces label noise that degrades the learned representations.

Figure 13 shows the three-panel comparison for the Experiment C chains, illustrating the improvement over the DESI pipeline baseline ($\sigma_{\rm robust} = 0.40$).

These results demonstrate that SPECVIT successfully transfers across three domain boundaries: (1) synthetic-to-real (BOSZ → APOGEE), (2) optical-to-infrared wavelength regime (710–885 nm → 1.51–1.70 μm), and (3) across instrumental configurations (PFS $R \sim 5000$ versus APOGEE $R \sim 22{,}500$). The ranking across experiments reveals a clear hierarchy: BOSZ-only priors (Exp A, $\sigma_{\rm robust} = 0.067$) > three-stage via DESI-50k (Exp C, $\sigma_{\rm robust} = 0.14$–$0.15$) > noisy DESI-360k pretrain (Exp B, $\sigma_{\rm robust} = 0.175$). More DESI data yields *worse* downstream performance, confirming that label noise in the DESI pipeline degrades the learned spectral representations.

The consistency between the two magnitude-regime Exp C models (both achieving $R^2 = 0.93$) suggests that performance is primarily determined by the quality of intermediate training labels rather than the noise model during BOSZ pretraining.

G. SYNTHETIC DATA GENERATION PIPELINE

This Appendix describes the procedure used to generate the synthetic stellar spectra dataset, including the BOSZ spectral library, parameter sampling, grid interpolation, and the PFS instrument noise model.

Table 17. Complete transfer learning results on APOGEE DR17 spectra.

Experiment	Training Chain	$\sigma_{\rm robust}$ (dex)	R^2	MAE (dex)	Test Set
A: SpecViT	BOSZ → APOGEE 7k	0.067	**0.954**	0.102	APOGEE 1k
A: BiLSTM	BOSZ → APOGEE 7k	**0.066**	0.953	**0.100**	APOGEE 1k
A: ConvStem+LSTM	BOSZ → APOGEE 7k	0.078	0.951	0.114	APOGEE 1k
B: Noisy Pretrain	BOSZ → DESI-360k → APOGEE 7k	0.175	0.882	0.183	APOGEE 1k
C: Three-stage (mag19–21)	BOSZ → DESI-50k → APOGEE 7k	0.14	0.93	···	APOGEE 1k
C: Three-stage (mag20.5–22.5)	BOSZ → DESI-50k → APOGEE 7k	0.15	0.93	···	APOGEE 1k
LightGBM baseline	APOGEE 7k only	0.111	0.953	0.122	APOGEE 1k
DESI pipeline	···	0.40	0.80	···	APOGEE 1k

Note—All test evaluations use the same 1,000-spectrum APOGEE test set ($T_{\rm eff} > 3600$ K). The robust sigma $\sigma_{\rm robust} = 1.4826 \times {\rm MAD}(\hat{y} - y)$ measures the robust scatter of residuals. Experiment A uses BOSZ-only pretraining with direct APOGEE fine-tuning (no DESI intermediate). Experiment B uses intermediate DESI-360k fine-tuning (APOGEE-excluded). Experiment C uses intermediate DESI-50k fine-tuning under two magnitude regimes.

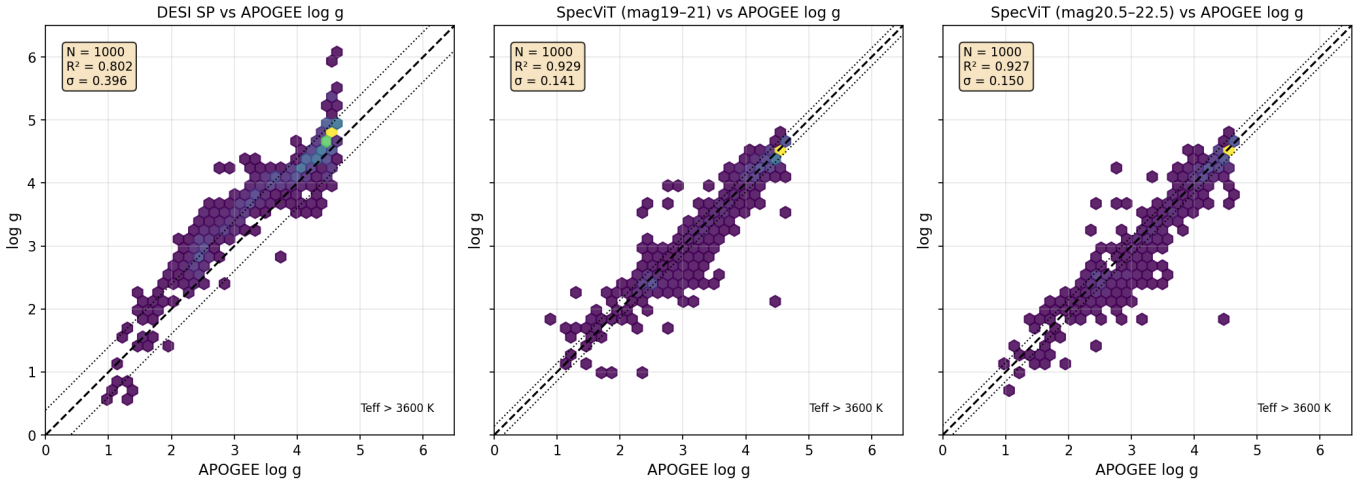

Figure 13. Comparison of $\log g$ estimates on the APOGEE–DESI crossmatched test set ($N = 1{,}000$, $T_{\rm eff} > 3600$ K) for the Experiment C (three-stage) pipeline. **Left:** DESI pipeline baseline ($\sigma_{\rm robust} = 0.40$ dex, $R^2 = 0.80$). **Middle:** SpecViT via BOSZ mag 19–21 → DESI-50k → APOGEE ($\sigma_{\rm robust} = 0.14$ dex, $R^2 = 0.93$). **Right:** Same pipeline with mag 20.5–22.5 pretraining ($\sigma_{\rm robust} = 0.15$ dex, $R^2 = 0.93$). The headline result (Experiment A: BOSZ → APOGEE, $\sigma_{\rm robust} = 0.067$, $R^2 = 0.954$) outperforms both three-stage variants, demonstrating that the DESI intermediate step is counterproductive. All panels show the 1:1 line (dashed) and $\pm 1\sigma_{\rm robust}$ envelopes (dotted).

G.1. *BOSZ Spectral Library*

The underlying spectral templates are drawn from the BOSZ (Bohlin–Osmer–Sahnow) grid (R. C. Bohlin et al. 2017), computed using ATLAS9 stellar atmospheres (F. Castelli & R. L. Kurucz 2004) with updated atomic and molecular line lists. The native resolution is $R \approx 50{,}000$, covering wavelengths from the ultraviolet through the near-infrared. For this work, we use the solar-scaled subset with metallicity [M/H] ranging from -2.5 to $+0.75$ dex in steps of 0.25 dex, effective temperature $T_{\rm eff}$ from 3500 to 12,000 K in varying steps (250 K at cool temperatures), and surface gravity $\log g$ from 0.0 to 5.0 dex in steps of 0.5 dex.

G.2. *Grid Interpolation*

To generate spectra at arbitrary stellar parameters within the BOSZ grid coverage, we employ cubic spline interpolation separately along each parameter axis. Specifically, for a target parameter set $(T_{\rm eff}, \log g, [{\rm M/H}])$:

1. Identify the enclosing hypercube of grid nodes.

2. Perform 1D cubic spline interpolation along each axis sequentially (temperature, then gravity, then metallicity).

3. Combine the interpolated fluxes to produce the final high-resolution template.

This approach preserves spectral line shapes and ensures smooth transitions across parameter space. The interpolated templates are then convolved with the instrument line-spread function (LSF) and resampled to the detector wavelength grid.

G.3. *PFS Instrument Model*

The Subaru Prime Focus Spectrograph (PFS) medium-resolution (MR) arm is modeled using the `pfsspec` simulation pipeline. Key instrument parameters are:

- **Wavelength coverage**: 710–885 nm.
- **Spectral resolution**: $R \approx 5000$ ($\Delta\lambda \approx 1.6$ Å).
- **Detector sampling**: 0.4 Å per pixel (4096 pixels total).
- **Line-spread function**: Gaussian with wavelength-dependent width derived from the optical model.

The convolution from native BOSZ resolution to PFS resolution is performed via a wavelength-dependent Gaussian kernel, followed by linear interpolation onto the fixed detector wavelength grid.

G.4. *Noise Model*

Observational noise is simulated by modeling the full photon-counting chain. For a spectrum with i-band apparent magnitude m_i and observing conditions (seeing, zenith angle, moon configuration), we compute:

1. **Object counts**: The expected photon count $N_{\mathrm{obj},j}$ at each wavelength pixel j is derived from the fluxed template, exposure time, and telescope/instrument throughput.

2. **Sky counts**: The sky background $N_{\mathrm{sky},j}$ is computed from a PFS sky model (new-moon conditions, dark time).

3. **Detector noise**: Read noise (approximately 3 e^- per pixel per read) and dark current are added.

4. **Total variance**: The per-pixel variance is $\sigma_j^2 = N_{\mathrm{obj},j} + N_{\mathrm{sky},j} + \sigma_{\mathrm{read}}^2$, following Poisson statistics for photon counts.

The noisy flux is generated by adding Gaussian noise with standard deviation σ_j to the noiseless (sky-subtracted, flux-calibrated) spectrum. The per-pixel error vector $\boldsymbol{\sigma}$ is stored alongside the flux and used during training for on-the-fly noise injection (Eq. 3).

G.5. *Dataset Partitioning*

The full dataset of 10^6 spectra is generated in five independent shards of 2×10^5 spectra each, using distinct random seeds for stellar parameter sampling and noise realization. An additional 10^3 spectra are generated for validation and 10^4 for testing. All three splits use disjoint stellar parameter draws and noise seeds to ensure no information leakage between training and evaluation.

H. TEMPLATE FITTING BASELINE

This Appendix describes the physics-based template fitting method used as a baseline in Table 3, which achieves $R^2 = 0.415$ for $\log g$ inference.

H.1. *Algorithm Overview*

Template fitting infers stellar parameters by maximizing the likelihood of the observed spectrum given a library of synthetic templates. For an observed spectrum $\boldsymbol{f}^{\rm obs}$ with per-pixel uncertainties $\boldsymbol{\sigma}$, the log-likelihood for a template $\boldsymbol{f}^{\rm mod}(\theta)$ at parameters $\theta = (T_{\rm eff}, \log g, [\mathrm{M/H}])$ is:

$$\ln \mathcal{L}(\theta) = -\frac{1}{2} \sum_{j=1}^{L} \frac{\left(f_j^{\rm obs} - A \cdot f_j^{\rm mod}(\theta)\right)^2}{\sigma_j^2}, \tag{H.1}$$

where A is a flux scaling factor (marginalized analytically) that accounts for distance, throughput, and continuum normalization. The best-fit parameters $\hat{\theta}$ are obtained by maximizing Eq. (H.1) over the BOSZ grid using spline-interpolated templates.

H.2. *Optimization Procedure*

The fitting proceeds in two stages:

1. **Coarse grid search**: Evaluate the log-likelihood on a subsampled grid (every second node in each parameter direction) to identify a promising region.

2. **Local refinement**: Use the Nelder–Mead simplex algorithm (J. A. Nelder & R. Mead 1965) starting from the coarse-grid optimum, with spline-interpolated templates evaluated at each function call.

The analytic flux normalization factor is computed at each grid point as:

$$A^* = \frac{\sum_j f_j^{\rm obs} f_j^{\rm mod} / \sigma_j^2}{\sum_j (f_j^{\rm mod})^2 / \sigma_j^2}, \tag{H.2}$$

which corresponds to the weighted least-squares solution for a linear scaling.

H.3. *Error Weighting*

The per-pixel error vector $\boldsymbol{\sigma}$ is used as inverse-variance weights ($w_j = 1/\sigma_j^2$). This naturally down-weights wavelength regions with high noise (e.g., sky emission lines, atmospheric absorption features, low-flux edges) and emphasizes regions with high signal-to-noise. The error vector encodes primarily *instrument and observing-condition* characteristics (detector response, sky brightness, exposure time) rather than information about the target spectrum itself, and therefore does not constitute data leakage.

H.4. *Implementation and Results*

Template fitting is implemented using the `pfsspec` stellar fitting module. On the full 10,000-spectrum test set (9,583 successful fits out of 9,600 attempts), the method achieves:

- $R^2(\log g) = 0.415$, MAE $= 0.67$ dex.
- $R^2(T_{\rm eff}) = 0.730$, MAE $= 247$ K.
- $R^2([\mathrm{M/H}]) = 0.855$, MAE $= 0.26$ dex.

The lower performance on $\log g$ compared to $T_{\rm eff}$ and metallicity reflects the weaker spectroscopic sensitivity of surface gravity: $\log g$ is encoded primarily in pressure-broadened line wings and subtle line-ratio diagnostics that are more easily overwhelmed by noise than the overall spectral shape (temperature) or line-depth modulation (metallicity).

I. BOOTSTRAP CONFIDENCE INTERVALS AND STATISTICAL TESTING

To quantify uncertainty in performance metrics and assess statistical significance, we employ bootstrap resampling on the test set.

Table 18. Bootstrap confidence intervals for main model comparison.

Model	R^2	95% CI	SE
SPECVIT	0.711	[0.699, 0.723]	0.0062
LightGBM	0.614	[0.602, 0.626]	0.0058
ΔR^2	0.097	[0.088, 0.106]	0.0046

NOTE—ΔR^2 is the improvement of SPECVIT over LightGBM. All intervals computed with $B = 1000$ bootstrap iterations.

I.1. *Bootstrap Procedure*

For each of the $B = 1000$ bootstrap iterations:

1. Draw $n = 10{,}000$ samples with replacement from the test set.
2. Compute R^2 for both SPECVIT and LightGBM on the bootstrap sample.
3. Record the difference $\Delta R^2 = R^2_{\text{ViT}} - R^2_{\text{LGBM}}$.

The 95% confidence interval is computed as the 2.5th and 97.5th percentiles of the bootstrap distribution. Standard errors are computed as the standard deviation of the bootstrap estimates.

I.2. *Statistical Significance Test*

To test whether SPECVIT significantly outperforms LightGBM, we compute the one-sided p-value as the fraction of bootstrap iterations where $\Delta R^2 \leq 0$:

$$p = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1} \left[\Delta R^2_b \leq 0 \right]. \tag{I.1}$$

In all 1000 bootstrap iterations, $\Delta R^2 > 0$, yielding $p < 0.001$. Combined with non-overlapping 95% confidence intervals (SPECVIT: [0.699, 0.723]; LightGBM: [0.602, 0.626]), this provides strong evidence that the performance improvement is statistically significant.

I.3. *Summary of Bootstrap Results*

J. FISHER INFORMATION COMPUTATION DETAILS

This Appendix provides implementation details for the Fisher-information ceiling presented in Section 5.1.4 and Appendix C.

J.1. *Regular Grid Requirement*

Reliable computation of the Fisher information matrix requires spectra evaluated on a *regular parameter grid* so that finite-difference derivatives can be computed consistently. For each magnitude/SNR bin, we generate a dedicated grid with structure:

- T_{eff}: 10 nodes spanning 3750–6000 K (250 K steps).
- $\log g$: 9 nodes spanning 1.0–5.0 dex (0.5 dex steps).
- [M/H]: 14 nodes spanning -0.25 to $+0.75$ dex (variable steps matching BOSZ grid).

This yields 1260 grid points per magnitude bin. For the 5D Fisher analysis, we additionally include [α/M] and [C/M] (2–3 nodes each), increasing the grid to approximately $10 \times 9 \times 14 \times 3 \times 3 = 11{,}340$ points.

J.2. *Jacobian Computation via Finite Differences*

At each interior grid point, we compute the spectral Jacobian $J = \partial f/\partial\theta$ using central finite differences along each parameter axis:

$$\frac{\partial f_j}{\partial \theta_k} \approx \frac{f_j(\theta_k + \Delta\theta_k) - f_j(\theta_k - \Delta\theta_k)}{2\Delta\theta_k}, \tag{J.1}$$

where $\Delta\theta_k$ is the grid spacing for parameter k. For a 3D parameter space $(T_{\rm eff}, \log g, [\mathrm{M/H}])$, this produces a $L \times 3$ Jacobian matrix at each grid point, where $L = 4096$ is the number of wavelength pixels.

Numerical stability.—Using a regular grid with consistent step sizes avoids the numerical instability encountered when using irregular (randomly sampled) parameter spacing. In early experiments, irregular grids produced CRLB values spanning 20 orders of magnitude due to inconsistent finite-difference denominators; the regular-grid approach reduces this range to approximately 3 orders of magnitude, consistent with physical expectations.

J.3. *Fisher Matrix and Marginalization*

Given the Jacobian J and the diagonal noise covariance $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_L^2)$, the Fisher information matrix is:

$$I(\theta) = J^\top \Sigma^{-1} J \in \mathbb{R}^{d\times d}, \tag{J.2}$$

where d is the number of parameters (3 for 3D, 5 for 5D). To obtain the CRLB for $\log g$ marginalized over nuisance parameters $\eta = (T_{\rm eff}, [\mathrm{M/H}], \ldots)$, we partition the Fisher matrix:

$$I = \begin{pmatrix} I_{gg} & I_{g\eta} \\ I_{\eta g} & I_{\eta\eta} \end{pmatrix}, \tag{J.3}$$

and compute the Schur complement:

$$\mathrm{CRLB}_{g,\mathrm{marg}} = \left(I_{gg} - I_{g\eta} I_{\eta\eta}^{-1} I_{\eta g}\right)^{-1}. \tag{J.4}$$

The Schur decay factor $\rho = (I_{gg} - I_{g\eta} I_{\eta\eta}^{-1} I_{\eta g})/I_{gg}$ quantifies the fraction of $\log g$ information retained after marginalizing over nuisance parameters; for our 3D (5D) analysis, $\rho \approx 0.69$ (0.58), indicating that approximately 30% (42%) of the raw $\log g$ information is "absorbed" by parameter degeneracy.

J.4. *CRLB to R^2 Conversion*

At each grid point, we convert the marginalized CRLB to a theoretical R^2 ceiling:

$$R^2_{\rm max} = 1 - \frac{\mathrm{CRLB}_{g,\mathrm{marg}}}{\mathrm{Var}(\log g)}, \tag{J.5}$$

where $\mathrm{Var}(\log g)$ is the label variance over the training distribution. The reported ceiling values in Table 13 are the *median* of $R^2_{\rm max}$ across all grid points at each magnitude, providing a representative summary. The 10th and 90th percentiles define the shaded confidence bands in Figure 1.

J.5. *5D vs 3D Ceiling Comparison*

Including chemical abundances ([α/M], [C/M]) as additional nuisance parameters reduces the $\log g$ ceiling by increasing parameter degeneracy. The effect is SNR-dependent:

- At high SNR (mag ≤ 20): $\Delta R^2_{\rm max} \lesssim 2\%$—chemical abundances are well-constrained and contribute little additional degeneracy.
- At low SNR (mag $= 22.5$): $\Delta R^2_{\rm max} \approx 28\%$—noise amplifies correlations between $\log g$ and abundance diagnostics.

For this reason, the main paper reports the 5D marginalized ceiling as a conservative upper bound on achievable performance.

K. VERIFICATION

This Appendix documents the verification procedures applied to ensure experimental rigor, including data integrity checks and protocol validation.

K.1. *No-Leakage Verification for DESI-50k Transfer Experiments*

To ensure that transfer learning results reported on the DESI-50k real-data benchmark are free of data leakage, we enforce and verify a strict split protocol at three levels: dataset construction, training configuration, and evaluation procedure.

K.1.1. *Split Disjointness (ID-Level Verification)*

Each HDF5 file in the DESI-50k dataset contains a `/dataset/params/targetid` column storing the unique DESI target identifier for each spectrum. We verify that these identifiers are *pairwise disjoint* across splits and contain *no within-split duplicates*:

$$|\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{val}}| = 0, \tag{K.1}$$

$$|\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}}| = 0, \tag{K.2}$$

$$|\mathcal{D}_{\text{val}} \cap \mathcal{D}_{\text{test}}| = 0, \tag{K.3}$$

where $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ denote the sets of target IDs in each split ($N_{\text{train}} = 40{,}000$, $N_{\text{val}} = 5{,}000$, $N_{\text{test}} = 5{,}000$). Additionally, all three splits contain zero duplicate IDs internally. This verification confirms that no test targets appear in the training or validation data.

K.1.2. *Training-Time Guardrails*

All fine-tuning runs are launched through a unified training script that constructs the experiment with `test_data=False`. Under this configuration, the data module does *not* instantiate a test dataloader, and the training loop executes only the fit stage (train/val) without accessing any test data.

As an additional hard guardrail specific to DESI-50k fine-tuning configurations, we explicitly set:

- `data.file_path` → `train/dataset.h5`
- `data.val_path` → `val/dataset.h5`
- `data.test_path` → `val/dataset.h5` (intentionally pointing to validation)

This configuration-level safeguard prevents any accidental reads of the held-out test file during training, even if a test dataloader were inadvertently triggered by downstream code.

K.1.3. *Held-Out Test Evaluation*

Final metrics reported on DESI-50k are computed *only after training completes* by running a standalone evaluation script on the held-out test split (`test/dataset.h5`). The evaluator is invoked with explicit arguments specifying:

- The checkpoint path (selected by best validation loss during training).
- The test file path (never seen during training).
- The normalization statistics source (`train/dataset.h5`), ensuring that label denormalization uses only training-set statistics.

This separation guarantees that test-set metrics reflect true generalization performance on data that was never used to fit model parameters or select hyperparameters.

K.2. *Canonical Data Specifications (Data Contract)*

To ensure fair comparison across all models—whether trained on synthetic BOSZ spectra or fine-tuned on real DESI data—we enforce a strict *data contract* that defines canonical dataset versions, immutable splits, and required data formats. This contract prohibits ad-hoc re-partitioning and guarantees that all reported metrics are computed on identical evaluation samples.

K.2.1. *DESI-50k Canonical Specification*

The real-data benchmark uses a single canonical dataset version with fixed splits:

- **Dataset name**: `DESI50k_BOSZstyle_matched_v1`
- **Representation**: BOSZ-style processed spectra on a 4096-pixel wavelength grid (matched to synthetic BOSZ data for consistent input dimensionality)
- **Target**: $\log g$ (dex)

The fixed splits are:

Split	N	Purpose
Train	40,000	Model fitting
Validation	5,000	Hyperparameter selection, early stopping
Test	5,000	Final evaluation (held out until paper-grade metrics)

Each HDF5 file must contain:

- `/dataset/arrays/flux/value`: float array of shape $(N, 4096)$
- `/dataset/params`: table with required columns `log_g` (float, dex) and `targetid` (unique identifier for leakage verification)

K.2.2. *BOSZ Synthetic Dataset Specification*

The synthetic training data uses BOSZ stellar atmosphere models (R. C. Bohlin et al. 2017) processed through the PFS instrument simulator:

- **Spectral library**: BOSZ (ATLAS9) at native resolution $R = 50{,}000$
- **Instrument model**: PFS Medium-Resolution (MR) arm, 4096 pixels
- **Wavelength coverage**: 710–885 nm
- **Parameter ranges**: $T_{\rm eff} \in [3750, 6000]$ K, $\log g \in [1.0, 5.0]$ dex, [M/H] $\in [-2.5, 0.75]$ dex
- **Magnitude range**: $i \in [19, 22.5]$ mag (regime-specific subsets as noted in experiments)
- **Observing conditions**: 3-hour exposure (12×900s), new-moon sky background

Training splits follow the standard partition: 10^6 train / 10^3 validation / 10^4 test, with disjoint random seeds for parameter sampling and noise realization.

K.2.3. *Immutability Rules*

The following hard rules are enforced by convention and automated checks:

1. **No re-splitting**: Once canonical splits are defined, they must not be modified. Creating new random splits for paper reporting is prohibited.
2. **No unsplit evaluation**: Evaluating on an unsplit "full pool" file (which could inadvertently mix training samples) is prohibited. All metrics must use the designated test split.
3. **Consistent input format**: All models (Ridge, LightGBM, SpecViT) receive identical input representations (4096-pixel flux vectors) from the same dataset files.
4. **Train-time isolation**: During training, only train and validation splits are accessed; the test split is reserved exclusively for final evaluation.

These constraints ensure that performance differences across models reflect algorithmic capability rather than data handling artifacts.

L. DESI TO BOSZ-STYLE PREPROCESSING PIPELINE

This Appendix documents the exact pipeline used to convert DESI DR1 spectra into BOSZ-style HDF5 files compatible with the synthetic BOSZ/PFS-MR training infrastructure. The canonical output dataset for paper benchmarks is `DESI50k_BOSZstyle_matched_v1` with fixed splits of $N = 40{,}000/5{,}000/5{,}000$ (train/val/test).

L.1. *Input Data Sources*

We support two construction paths depending on data availability:

Path 1: Local H5 (flux only).—When DESI spectra have been pre-downloaded to a local HDF5 file containing rest-framed flux arrays (shape $N \times 7781$) with associated stellar parameters (`teff`, `logg`, `feh`) and target identifiers (`targetid`), we use these directly. This path produces a constant placeholder for per-pixel errors (see §L.6).

Path 2: SPARCL re-fetch (flux + ivar).—Alternatively, we query the SPARCL archive by `targetid` to obtain observed-frame wavelength, flux, inverse-variance (`ivar`), and mask arrays. This path enables computation of physically meaningful per-pixel uncertainties via $\sigma = 1/\sqrt{\text{ivar}}$.

L.1.1. *Data Provenance*

DESI DR1 release and citation. Spectra and stellar parameters used in this work come from the Dark Energy Spectroscopic Instrument (DESI) Data Release 1 (DESI Collaboration 2025). The survey design and targeting are described in DESI Collaboration (2016). The official data archive is https://data.desi.lbl.gov/public/dr1; when using DESI DR1 data, the collaboration requests citation of the DR1 paper and the acknowledgments at https://data.desi.lbl.gov/doc/acknowledgments.

SPARCL (NOIRLab spectral archive). SPARCL (SPectra Analysis and Retrievable Catalog Lab) is NOIRLab's spectral archive service, providing programmatic access to DESI and other survey spectra without direct access to the raw coadd files. The Python client is `sparclclient` (https://pypi.org/project/sparclclient/); install with `pip install sparclclient`. After loading a table of `targetid` values (and optional stellar parameters), spectra are retrieved in batches via the client's `retrieve_by_specid` method with `dataset_list=["DESI-DR1"]` and `include=["wavelength","flux","ivar","mask","targetid","redshift",...]`, returning observed-frame wavelength, flux, inverse-variance, and mask arrays per target.

MWS target selection. The Milky Way Survey (MWS) catalog and stellar parameters (e.g., `teff`, `logg`, `feh`, `vrad`) are taken from the DESI DR1 MWS/iron value-added catalog: https://data.desi.lbl.gov/public/dr1/vac/dr1/mws/iron/v1.0/ (e.g., `rvpix-main-bright.fits` in `rv_output`). Selection criteria used to build the initial 50k subset align with the BOSZ parameter ranges in §L.2: $T_{\rm eff} \in [3750, 6000]$ K, $\log g \in [1.0, 5.0]$ dex, [Fe/H] $\in [-2.5, 0.75]$ dex, with optional SNR and quality cuts. A typical workflow is: (1) download the MWS catalog FITS to a local path; (2) select a random or stratified subset of `targetid`s satisfying the above cuts and export to a CSV with columns `targetid`, `teff`, `logg`, `feh` (and `vrad` if available); (3) fetch spectra via SPARCL using that CSV.

Initial 50k H5 build (fetch command example). The pre-downloaded local H5 used in Path 1 is produced by: (1) downloading the MWS catalog (e.g., `rvpix-main-bright.fits`) from the archive URL above; (2) running a selection script to output a CSV of 50,000 `targetid`s (and parameters) within the BOSZ-like ranges; (3) running the SPARCL fetch script with that CSV, a target wavelength grid (e.g., 7781 pixels), and an output HDF5 path, e.g.:

```
fetch_desi_spectra_sparcl.py --subset-csv <path/to/desi_subset_50k.csv> --wavelength-txt <path/to/wavelength.txt>
--out-h5 <path/to/desi_test_50k_vrad.h5> --batch 200 --restframe auto
```

The script uses `SparclClient().retrieve_by_specid(...)` in batches (default 200), applies rest-frame correction using `vrad` or SPARCL redshift, interpolates onto the given wavelength grid, and writes `test/spectra`, `test/targetid`, `test/teff`, `test/logg`, `test/feh`. Downstream BOSZ-style conversion (resampling to 4096 pixels, median normalization, spike masking) then produces the canonical `DESI50k_BOSZstyle_matched_v1` splits.

L.2. *Parameter-Range Filtering*

Before splitting, we filter DESI targets to align with the BOSZ low-temperature training regime:

$$T_{\rm eff} \in [3750, 6000] \text{ K}, \tag{L.1}$$

$$\log g \in [1.0, 5.0] \text{ dex}, \tag{L.2}$$

$$[\text{Fe/H}] \in [-2.5, 0.75] \text{ dex}. \tag{L.3}$$

This hard filter reduces out-of-domain extrapolation when applying BOSZ-trained models to real data.

L.3. *Rest-Frame Correction*

For SPARCL-based construction, observed wavelengths are converted to rest-frame using either the radial velocity $v_{\rm rad}$ (if available):

$$\lambda_{\rm rest} = \frac{\lambda_{\rm obs}}{1 + v_{\rm rad}/c}, \tag{L.4}$$

or the SPARCL redshift z:

$$\lambda_{\rm rest} = \frac{\lambda_{\rm obs}}{1 + z}. \tag{L.5}$$

For the local H5 path, spectra are assumed to be pre-corrected to rest-frame.

L.4. *Resampling to BOSZ Wavelength Grid*

Each spectrum is interpolated onto the BOSZ 4096-pixel wavelength grid (710–885 nm, matching the PFS-MR arm) using linear interpolation with explicit fill values:

- **Flux**: pixels outside the source wavelength coverage or with non-finite values are filled with 0.5 (the median-normalized continuum level).
- **Error**: out-of-range pixels are filled with $+\infty$ (effectively zero weight in downstream analysis).

After interpolation, flux values are clipped to be non-negative.

L.5. *Spike Masking*

To remove cosmic-ray artifacts and instrumental spikes, we apply a robust smooth-baseline replacement:

1. Compute a moving-average baseline s with window length $w = 101$ pixels.
2. Compute residuals $r = f - s$.
3. Estimate robust scatter via the median absolute deviation: $\hat{\sigma} = 1.4826 \cdot \mathrm{MAD}(r)$.
4. Flag pixels where $|r - \mathrm{median}(r)| > k\hat{\sigma}$ with $k = 10$.
5. Replace flagged pixels with the baseline value: $f_{\rm out}[\mathrm{spike}] \leftarrow s[\mathrm{spike}]$.

This conservative threshold ($k = 10$) preserves real spectral features while removing outliers.

L.6. *Median Normalization*

To align with BOSZ preprocessing conventions, each spectrum undergoes per-spectrum median normalization:

$$f' = f \cdot \frac{0.5}{\mathrm{median}(f_{f>0})}, \tag{L.6}$$

where the median is computed over positive finite pixels. This maps the continuum level to approximately 0.5.

Error scaling (SPARCL path).—When per-pixel σ is available, it is scaled by the same factor:

$$\sigma' = \sigma \cdot \frac{0.5}{\mathrm{median}(f_{f>0})}. \tag{L.7}$$

Error placeholder (local H5 path).—When no inverse-variance exists, we write a constant placeholder $\sigma = 0.02$ for all pixels. This enables consumers that require the error array to exist; Ridge and LightGBM baselines use flux only and ignore this field.

L.7. *SNR Estimation*

Two SNR estimators are implemented depending on the construction path:

Robust second-difference estimator (local H5 path).—When per-pixel errors are unavailable, we estimate noise from the flux array using the robust MAD of the second difference:

$$\hat{\sigma}_{\text{noise}} = \frac{1.4826}{\sqrt{6}} \cdot \text{median}\,|2f_i - f_{i-1} - f_{i+1}|\,, \tag{L.8}$$

$$\text{SNR} = \frac{\text{median}(f)}{\hat{\sigma}_{\text{noise}}}. \tag{L.9}$$

This estimator is computed *before* median normalization to avoid scale inflation.

Direct estimator (SPARCL path).—When per-pixel σ is available:

$$\text{SNR} = \text{median}\left(\frac{f_i}{\sigma_i}\right) \tag{L.10}$$

over finite pixels.

L.8. *Deterministic Train/Val/Test Split*

After filtering (and optional downsampling), we create splits deterministically:

1. Shuffle indices with fixed random seed (default: 42).
2. Allocate fractions: $f_{\text{val}} = 0.1$, $f_{\text{test}} = 0.1$.
3. Assign remaining samples to training.

For $N = 50{,}000$ filtered samples, this yields the canonical split sizes: 40,000 / 5,000 / 5,000.

L.9. *Output HDF5 Schema*

Each split produces one file `<split>/dataset.h5` containing:

- `spectrumdataset/wave`: shape $(4096,)$, float32—the BOSZ wavelength grid.
- `dataset/arrays/flux/value`: shape $(N, 4096)$, float32, gzip-compressed.
- `dataset/arrays/error/value`: shape $(N, 4096)$, float32, gzip-compressed.
- `/dataset/params`: pandas HDF table containing:
  - `targetid` (int): unique DESI identifier for leakage verification.
  - `log_g` (float): surface gravity label in dex.
  - `T_eff`, `Fe_H` (float): stellar parameters.
  - `snr` (float): estimated signal-to-noise ratio.
  - Optional: `redshift`, `vrad`, `mag`.

This schema is directly compatible with the BOSZ training infrastructure, enabling seamless transfer learning experiments.

REFERENCES

Abdurro'uf, Accetta, K., Aerts, C., et al. 2022, The Astrophysical Journal Supplement Series, 259, 35, doi: 10.3847/1538-4365/ac4414

Allende Prieto, C. 2016, FERRE User's Guide,, Technical report

Allende Prieto, C., Beers, T. C., Wilhelm, R., et al. 2006, The Astrophysical Journal, 636, 804, doi: 10.1086/498131

Blanco-Cuaresma, S., Soubiran, C., Heiter, U., & Jofré, P. 2014, Astronomy & Astrophysics, 569, A111

Bohlin, R. C., Mészáros, S., Fleming, S. W., Gordon, K. D., & Koesterke, L. 2017, The Astronomical Journal, 153, 234
Breiman, L. 2001, Machine Learning, 45, 5, doi: 10.1023/A:1010933404324
Castelli, F., & Kurucz, R. L. 2004, New Grids of ATLAS9 Model Atmospheres, doi: 10.48550/arXiv.astro-ph/0405087
Cortes, C., & Vapnik, V. 1995, Machine Learning, 20, 273, doi: 10.1007/BF00994018
Cramér, H. 1946, Mathematical Methods of Statistics (Princeton University Press)
Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, Research in Astronomy and Astrophysics, 12, 1197
DESI Collaboration. 2016, The DESI Experiment Part I: Science,Targeting, and Survey Design, doi: 10.48550/arXiv.1611.00036
DESI Collaboration. 2025, arXiv e-prints, doi: 10.48550/arXiv.2503.14745
Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2021, in International Conference on Learning Representations (ICLR). https://arxiv.org/abs/2010.11929
Fabbro, S., Venn, K. A., O'Briain, T., Bialek, S., & other. 2018, Monthly Notices of the Royal Astronomical Society, 475, 2978
Falcon, W., & The PyTorch Lightning team. 2019, PyTorch Lightning,, https://github.com/Lightning-AI/lightning doi: 10.5281/zenodo.3828935
Fisher, R. A. 1925, Mathematical Proceedings of the Cambridge Philosophical Society, 22, 700
Friedman, J. H. 2001, The Annals of Statistics, 29, 1189
Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., et al. 2016, Astronomy & Astrophysics, 595, A1, doi: 10.1051/0004-6361/201629272
García Pérez, A. E., Allende Prieto, C., Holtzman, J. A., et al. 2016, The Astronomical Journal, 151, 144
Gustafsson, B., Edvardsson, B., Eriksson, K., et al. 2008, Astronomy & Astrophysics, 486, 951
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778
Hoerl, A. E., & Kennard, R. W. 1970, Technometrics, 12, 55, doi: 10.1080/00401706.1970.10488634
Hoffmann, J., Borgeaud, S., Mensch, A., et al. 2022, Training Compute-Optimal Large Language Models, https://arxiv.org/abs/2203.15556
Husser, T.-O., Wende-von Berg, S., Dreizler, S., et al. 2013, Astronomy & Astrophysics, 553, A6, doi: 10.1051/0004-6361/201219058
Islam, M. K., & Fox, J. 2026, OmniSpectra: A Unified Foundation Model for Native Resolution Astronomical Spectra, https://arxiv.org/abs/2601.15351
Kaplan, J., McCandlish, S., Henighan, T., et al. 2020, Scaling Laws for Neural Language Models, https://arxiv.org/abs/2001.08361
Karniadakis, G. E., Kevrekidis, I. G., Lu, L., et al. 2021, Nature Reviews Physics, 3, 422, doi: 10.1038/s42254-021-00314-5
Kay, S. M. 1993, Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (Prentice Hall)
Ke, G., Meng, Q., Finley, T., et al. 2017, in Advances in Neural Information Processing Systems
Koblischke, N., & Bovy, J. 2024, SpectraFM: Tuning into Stellar Foundation Models, doi: 10.48550/arXiv.2411.04750
Lee, Y. S., Beers, T. C., Sivarani, T., et al. 2008, The Astronomical Journal, 136, 2022
Leung, H. W., & Bovy, J. 2019, Monthly Notices of the Royal Astronomical Society, 483, 3255
Li, J., et al. 2023, MNRAS, 521, 6354
Li, Z., et al. 2025, Stellar Atmospheric Parameter Estimation from CSST Slitless Spectra Using a Fully Connected Residual Network, https://arxiv.org/abs/2512.10345
Liu, Z., Lin, Y., Cao, Y., et al. 2021, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). https://arxiv.org/abs/2103.14030
Moraes, C. M., Campos, C. A., Figueiredo, E. G. T., & Cerqueira, E. D. B. 2025, Applying Vision Transformers on Spectral Analysis of Astronomical Objects, doi: 10.48550/arXiv.2506.00294
Nelder, J. A., & Mead, R. 1965, The Computer Journal, 7, 308, doi: 10.1093/comjnl/7.4.308
Ness, M., Hogg, D. W., Rix, H.-W., Ho, A. Y. Q., & Zasowski, G. 2015, The Astrophysical Journal, 808, 16
Pál, B., Dobos, L., & Budavári, T. 2024, arXiv preprint arXiv:2409.11625. https://arxiv.org/abs/2409.11625
Paszke, A., Gross, S., Massa, F., et al. 2019, in Advances in Neural Information Processing Systems, Vol. 32
Pattnaik, R., Kartaltepe, J. S., & Binu, C. 2025, SpecPT (Spectroscopy Pre-trained Transformer) Model for Extragalactic Spectroscopy: I. Architecture and Automated Redshift Measurement, doi: 10.48550/arXiv.2501.01070
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Press, O., Smith, N. A., & Lewis, M. 2021, Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation, https://arxiv.org/abs/2108.12409

Raissi, M., Perdikaris, P., & Karniadakis, G. E. 2019, Journal of Computational Physics, 378, 686

Rao, C. R. 1945, Bulletin of the Calcutta Mathematical Society, 37, 81

Rozański, T., & Ting, Y.-S. 2025, Scaling Laws for Stellar Spectral Emulation, https://arxiv.org/abs/2503.18617

Rozański, T., Ting, Y.-S., & Jabłońska, M. 2025, ApJ, 980, 66

Shaw, P., Uszkoreit, J., & Vaswani, A. 2018, in Proceedings of NAACL-HLT. https://arxiv.org/abs/1803.02155

Su, J., Lu, Y., Pan, S., et al. 2021, RoFormer: Enhanced Transformer with Rotary Position Embedding, https://arxiv.org/abs/2104.09864

Tamura, N., Takato, N., Shimono, A., et al. 2016, in Ground-based and Airborne Instrumentation for Astronomy VI (Proc. SPIE), Vol. 9908

Tegmark, M., Taylor, A. N., & Heavens, A. F. 1997, The Astrophysical Journal, 480, 22

Ting, Y.-S., Conroy, C., & Rix, H.-W. 2019, The Astrophysical Journal, 879, 69

Touvron, H., Cord, M., Douze, M., et al. 2021, in Proceedings of the International Conference on Machine Learning (ICML). https://arxiv.org/abs/2012.12877

Valenti, J. A., & Piskunov, N. 1996, Astronomy and Astrophysics Supplement Series, 118, 595

Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, in Advances in Neural Information Processing Systems. https://arxiv.org/abs/1706.03762

Wu, Y., Luo, A., Du, B., Zhao, Y., & Yuan, H. 2014, Automatic stellar spectral parameterization pipeline for LAMOST survey, doi: 10.48550/arXiv.1407.1980

Xiang, M., Liu, X., Yuan, H.-L., et al. 2015, Monthly Notices of the Royal Astronomical Society, 448, 822

Yang, Y., & Li, X. 2024, Universe, 10, 214, doi: 10.3390/universe10050214

York, D. G., Adelman, J., Anderson, J. E., et al. 2000, The Astronomical Journal, 120, 1579, doi: 10.1086/301513

Zhao, X., Li, X., Li, H., & Liu, X. 2025, The Astrophysical Journal Supplement Series, 278, 41, doi: 10.3847/1538-4365/adcf9b