# Learning to Look: Cognitive Attention Alignment with Vision-Language Models

**Ryan L. Yang***
Brown University
Providence, RI
ryan_l_yang@brown.edu

**Dipkamal Bhusal***
Rochester Institute of Technology
Rochester, NY
db1702@rit.edu

**Nidhi Rastogi**
Rochester Institute of Technology
Rochester, NY
nxrvse@rit.edu

## Abstract

Convolutional Neural Networks (CNNs) frequently "cheat" by exploiting superficial correlations, raising concerns about whether they make predictions for the right reasons. Inspired by cognitive science, which highlights the role of attention in robust human perception, recent methods have sought to guide model attention using concept-based supervision and explanation regularization. However, these techniques depend on labor-intensive, expert-provided annotations, limiting their scalability. We propose a scalable framework that leverages vision-language models to automatically generate semantic attention maps using natural language prompts. By introducing an auxiliary loss that aligns CNN attention with these language-guided maps, our approach promotes more reliable and cognitively plausible decision-making without manual annotation. Experiments on challenging datasets, ColoredMNIST and DecoyMNIST, show that our method achieves state-of-the-art performance on ColorMNIST and remains competitive with annotation-heavy baselines on DecoyMNIST, demonstrating improved generalization, reduced shortcut reliance, and model attention that better reflects human intuition. Our code is available at https://github.com/ryanlyang/LearningToLook/.

## 1 Introduction

Despite their impressive performance, Convolutional Neural Networks (CNNs) tend to "cheat" during learning, exploiting superficial correlations and spurious features in data rather than acquiring robust, generalizable representations [3]. This shortcut learning raises a critical question for building reliable AI systems: *Are models right for the right reasons?*.

Cognitive science offers a rich tradition of studying not only what intelligence can do, but how it achieves its capabilities. Human perception, for example, is guided by attention mechanisms that flexibly allocate cognitive resources to task-relevant features. One of the simplest examples is how we identify objects by their distinctive shapes, patterns or colors, ignoring irrelevant distractions. These mechanisms support robust generalization and interpretable reasoning. Inspired by such insights, recent efforts have influenced the strategies for attention and feature selection in neural networks [9, 8, 4].

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models.

One of the popular approaches is to guide models toward human-meaningful, task-relevant regions when making decisions. Prior work has introduced concept-based supervision [4] and explanation regularization techniques [9, 8], where models are encouraged to align their saliency or concept activations with expert-provided ground-truth annotations through auxiliary losses. However, these approaches require manual collection of ground truth for saliency or, dataset for concepts with deep domain expertise, and can introduce annotation biases, restricting their scalability and applicability.

In this work, we present a scalable framework that leverages advances in vision-language models to automate cognitively meaningful attention supervision. Specifically, we utilize WeCLIP+ [11] to generate attention maps for arbitrary visual concepts using natural language prompts, serving as "teacher" signals grounded in semantics. During training, we introduce an auxiliary loss that aligns the model's attention with these language-guided reference maps, steering CNNs toward more reliable, and cognitively plausible decision-making, without the bottleneck of manual annotation.

We seek to explore the following research questions: *Can we use language and vision to instill cognitively motivated inductive biases in neural networks? Does this reduce shortcut reliance and improve robustness or generalization?* We investigate these questions on challenging classification tasks such as ColoredMNIST [7] and DecoyMNIST [2]. Our method achieves state-of-the-art results on ColoredMNIST and remains competitive with annotation-intensive baselines on DecoyMNIST, despite relying only on automatically generated pseudo-maps.

## 2 Related work

Concept Distillation [4] utilizes the concept activation vectors (CAVs) framework from TCAV [5] by introducing a concept loss to fine-tune models for reducing bias and improving alignment with human-understood concepts. However, such methods rely on manually defined and collected concept samples, which are costly to annotate and may introduce biases reflective of the provided dataset. Another approach, CDEP [8], enables the integration of domain knowledge by penalizing models whose explanations do not align with expert-identified, task-relevant features. However, it still requires extensive domain expertise and human annotation to construct ground-truth explanations, limiting its scalability. Similarly, Right for the Right Reasons [9] introduces an input gradient regularization technique, where model training is guided by selectively penalizing input gradients corresponding to features identified (by experts) as irrelevant or spurious. While effective in encouraging models to base decisions on the "right" features, this approach also depends on detailed, expert-provided annotations to specify which features should or should not influence model predictions.

## 3 Methodology

Our proposed framework consists of two principal stages: (1) generating class and concept-specific attention maps using a vision-language model, and (2) training a CNN with an auxiliary loss that aligns its attention with these automatically generated maps. In this section, we formally introduce our notation and describe each component in detail.

### 3.1 Preliminaries and Notation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a dataset of $N$ images $x_i \in \mathbb{R}^{H \times W \times 3}$ with corresponding class labels $y_i \in \mathcal{C}$, where $\mathcal{C}$ denotes the set of all possible classes. We denote our CNN model as $f_\theta$, parameterized by weights $\theta$, which outputs logits $f_\theta(x)$ for input $x$.

For an input $x$ and class $c$, a *saliency map* $S_\theta(x, c) \in [0, 1]^{H \times W}$ visualizes the importance of each pixel in $x$ for predicting class $c$, typically obtained using a method such as IG [10] or Class Activation Mapping (CAM)[12]. Similarly, $M_{\mathrm{VL}}(x, c) \in [0, 1]^{H \times W}$ denotes an attention map for concept $c$ in $x$ generated by a vision-language model.

### 3.2 Automatic Attention Map Generation

To obtain supervision signals without manual annotation, we employ WeCLIP+[11], a state-of-the-art vision-language model, to generate class-specific attention maps. For each $(x_i, y_i)$, we construct a natural language prompt $t_{y_i}$ (e.g., "a photo of a digit") that matches the class semantics. Optionally,

we also provide background or distractor prompts to help the model distinguish target concepts from context. In Appendix B, we discuss the prompts for our targeted classification tasks.

WeCLIP+ computes an affinity map $M_{\mathrm{VL}}(x_i, y_i)$ that highlights image regions associated with the semantic concept $y_i$ and the natural language prompt. These maps are automatically generated for all images in the training and validation sets and used as pseudo ground-truth for attention alignment.

**Attention Map Post-processing.** To improve alignment with desired inductive biases, we optionally refine the raw attention maps. For instance, morphological dilation with a structuring element of radius $r$ can be applied to ensure that the entire object is covered, or edge detection (e.g., Canny operator followed by dilation [1]) can be used to focus the model on object boundaries. Such preprocessing is task-dependent and controlled via simple hyperparameters. However, we observed that the unmodified attention maps produced by WeCLIP+ often perform well enough on their own.

### 3.3 Attention-Aligned CNN Training

Given the dataset $\mathcal{D}$ and the set of generated attention maps $\{M_{\mathrm{VL}}(x_i, y_i)\}$, we train the CNN $f_\theta$ to simultaneously minimize classification error and encourage its internal attention to match the WeCLIP+ pseudo-masks.

**Classification Loss.** For each mini-batch, the standard cross-entropy loss is computed as:

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{B}\sum_{i=1}^{B} \log p_\theta(y_i \mid x_i), \tag{1}$$

where $B$ is the batch size and $p_\theta(y_i \mid x_i)$ is the softmax probability of the correct class.

**Attention Alignment Loss.** For each sample, we compute a normalized saliency map $S_\theta(x_i, y_i)$ for the true label $y_i$ using class activation mapping (CAM). We then minimize the Kullback–Leibler (KL) divergence between this map and the WeCLIP+ attention map, both normalized so that $\sum_{h,w} S_\theta(x_i, y_i)[h, w] = 1$ and similarly for $M_{\mathrm{VL}}(x_i, y_i)$:

$$\mathcal{L}_{\mathrm{attn}} = \frac{1}{B}\sum_{i=1}^{B} \mathrm{KL}\big(S_\theta(x_i, y_i) \parallel M_{\mathrm{VL}}(x_i, y_i)\big). \tag{2}$$

The model is optimized with a weighted combination of classification and attention alignment losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\, \mathcal{L}_{\mathrm{attn}}, \tag{3}$$

where $\lambda > 0$ determines the strength of the attention supervision. We train with a two-phase schedule: for the first $E_{\mathrm{attn}}$ epochs we optimize only $\mathcal{L}_{\mathrm{attn}}$ to "learn to look"; at $E_{\mathrm{attn}}$ we reset the optimizer/scheduler and thereafter minimize the joint objective in Eqn. 3.

## 4 Experiments

We empirically evaluate our language-guided attention alignment framework on two challenging biased classification benchmarks: ColorMNIST [7] and DecoyMNIST [2]. These datasets are designed to test a model's reliance on spurious correlations and its ability to generalize beyond shortcut cues.

### 4.1 Setup

**Datasets.** **ColorMNIST** [7] is a variant of MNIST [6] in which each digit class is assigned a unique color in the training set. At test time, the color mapping is reversed, forcing models to rely on digit shape rather than color for accurate classification. In **DecoyMNIST** [2], MNIST digits are augmented by adding class-indicative gray patches to the image boundary. These "decoy" patches create a spurious association between digit class and patch location or intensity.

**Baselines and Comparison.** We compare our approach (**Ours**) to a baseline CNN (**Base**), concept distillation based supervision (CDBS) [4], and explanation regularization techniques, right-for-right-reasons (RRR) [9] and penalizing explanations (CDEP) [8].

**Training Details.** We train a LeNet [6] with SGD (momentum $0.98$, weight decay $10^{-4}$), batch size 32, for 30 epochs. The only dataset-specific hyperparameter is the initial learning rate: $10^{-3}$ for ColorMNIST and $10^{-2}$ for DecoyMNIST, decayed by a factor of $0.1$ every 7 epochs; all other settings are identical.

Training runs in two phases separated by the *Attention epoch* $E_{\text{attn}}$. Phase 1 ("learn to look"): optimize only $\mathcal{L}_{\text{attn}}$ (KL between the model CAM and the WeCLIP+ pseudo-map). At $E_{\text{attn}}$ we reset the optimizer and scheduler (zero momentum, restart LR) while keeping network weights unchanged. Phase 2: minimize $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda\,\mathcal{L}_{\text{attn}}$ with a ramp on $\lambda$: for each epoch $e \geq E_{\text{attn}}$, $\lambda_{e+1} = \lambda_e + 0.1\,\lambda_0$ (with $\lambda_{E_{\text{attn}}} = \lambda_0$) to keep attention prioritized.

We tune $\lambda$ and $E_{\text{attn}}$ on a validation set using a composite metric we call *Optim Value*:

$$\text{Optim Value} \;=\; \text{ValAcc} \times \big(1 - \mathcal{L}_{\text{attn}}\big),$$

favoring configurations that jointly increase accuracy and decrease attention divergence.

See Appendix C for the full $(\lambda, E_{\text{attn}})$ grid search and selection criterion; heatmaps are shown in Fig. 2. We select $\lambda$=160, $E_{\text{attn}}$=11 for ColorMNIST and $\lambda$=8, $E_{\text{attn}}$=13 for DecoyMNIST

**Mask preprocessing (ColorMNIST only).** To bias the model toward *shape* rather than color on ColorMNIST, we preprocess the WeCLIP+ pseudo-masks before computing $\mathcal{L}_{\text{attn}}$. Concretely, we first apply a small morphological dilation to ensure the digit is fully covered, then extract a thin boundary band from the dilated mask (edge detection followed by a light dilation). For DecoyMNIST, we use the raw pseudo-masks without preprocessing.

### 4.2 Result

Table 1: Test accuracy (%) on biased benchmarks. Higher is better. mean ± s.d. over 5 random seeds.

| | **Base** | **CDEP**[8] | **RRR** [9] | **CDBS** [4] | **Ours** |
|---|---|---|---|---|---|
| **ColoredMNIST** | 0.1 | 31.0 | 0.1 | 50.93 | 64.88 ± 2.85 |
| **DecoyMNIST** | 52.8 | 97.2 | 99.0 | 98.9 | 96.19 ± 0.35 |

Table 1 reports test accuracy on the biased dataset benchmarks. On **ColoredMNIST**, the baseline CNN (**Base**) fails completely, achieving only $0.1\%$ accuracy, confirming it has learned to classify digits entirely by color shortcuts. The explanation-regularization method **RRR** [9] also performs poorly in this setting, while textbfCDEP [8] improves generalization to $31.0\%$ accuracy, and the concept-distillation approach **CDBS** [4] reaches $50.93\%$. Our method achieves the best result with $64.88 \pm 2.85\%$, demonstrating that language-guided attention alignment can reduce shortcut reliance more effectively than annotation-heavy baselines. Qualitative maps in Appendix A illustrate that our method shifts saliency from background color to digit shape.

On **DecoyMNIST**, the baseline attains $52.8\%$, again showing substantial reliance on the corner-patch shortcut. Manual supervision methods perform near-perfectly, with **RRR** reaching $99.0\%$, **CDEP** $97.2\%$, and **CDBS** $98.9\%$. Our method achieves $96.19 \pm 0.35\%$, slightly below the annotation-heavy approaches but still competitive, while requiring no human-provided saliency or concept labels. Qualitative results (Appendix A) confirm that our model attends primarily to the digit body rather than patch artifacts. shifts.

## 5 Limitations

Our work has few limitations that open avenues for future research. First, attention maps are precomputed and stored, which can be memory-intensive. In future work, we plan to explore on-the-fly generation techniques. Second, the current evaluation is restricted to relatively simple datasets (ColoredMNIST and DecoyMNIST), and future work will investigate performance on more complex benchmarks to assess broader generalizability. Finally, reliance on a vision-language model as an attention teacher may introduce its own biases. Future work will investigate strategies to mitigate such risks.

## 6 Conclusion

We presented a scalable, annotation-free framework that leverages language-driven attention maps from vision-language models to guide neural networks toward task-relevant, human-meaningful features. Empirically, our approach achieves state-of-the-art accuracy on ColorMNIST and remains competitive with annotation-intensive baselines on DecoyMNIST, despite requiring no human-provided saliency maps or concept sets. Our framework is backbone-agnostic, supporting a variety of CNNs and differentiable saliency techniques, and can be extended to architectures such as Vision Transformers by adapting the attribution mechanism.


## Acknowledgement

R. Yang and N. Rastogi were partially supported by NSF Award #2447631.


## References


[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 2009.

[2] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

[3] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[4] Avani Gupta, Saurabh Saini, and PJ Narayanan. Concept distillation: leveraging human-centered explanations for model improvement. *Advances in Neural Information Processing Systems*, 36: 63724–63737, 2023.

[5] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[7] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019.

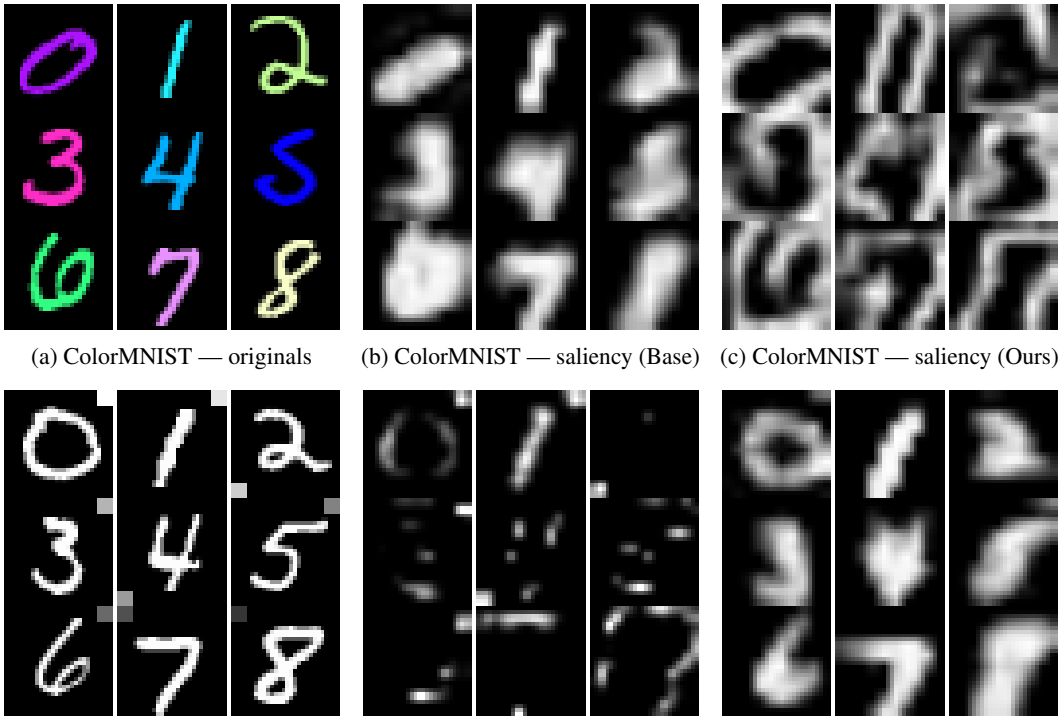
[8] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020.

[9] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[10] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[11] Bingfeng Zhang, Siyue Yu, Jimin Xiao, Yunchao Wei, and Yao Zhao. Frozen clip-dino: A strong backbone for weakly supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.



(a) ColorMNIST — originals (b) ColorMNIST — saliency (Base) (c) ColorMNIST — saliency (Ours)

(d) DecoyMNIST — originals (e) DecoyMNIST — saliency (Base) (f) DecoyMNIST — saliency (Ours)

Figure 1: **Qualitative comparison by dataset.** Each row shows 3×3 grids of *(left)* original inputs, *(middle)* saliency without attention alignment, and *(right)* saliency with attention alignment. Brighter regions indicate higher saliency (e.g., CAM).

## A Saliency Map Comparison on ColorMNIST and DecoyMNIST

Figure 1 demonstrates the saliency comparison using CAM before and after attention alignment.

## B Prompts for Targeted Classification Tasks

We use a single foreground class (`digit`) for both datasets to emphasize shape over spurious cues. Prompts are short, generic, and stable across images.

### B.1 ColoredMNIST

**Foreground (class) prompts.**

- `digit`

**Background categories.**

- `Background`, `dark`, `black`

### B.2 DecoyMNIST

**Foreground (class) prompts.**

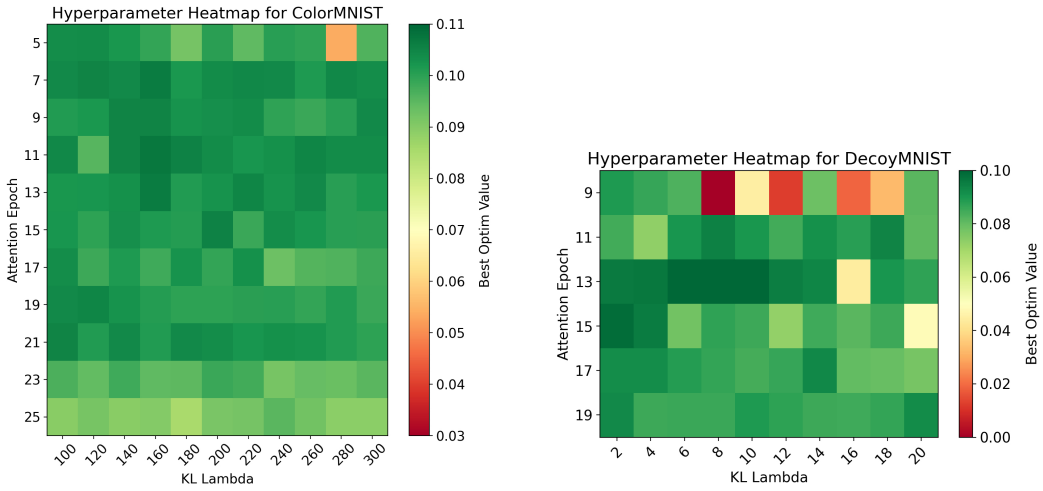
- `digit`

**Background categories.**

- `Background`, `dark`, `black`, `corner`, `patch`, `box`, `corner patch`

*Rationale.* We use a single neutral noun (`digit`) to keep the prompt stable and avoid leaking color or style cues, while the background tokens name the dominant nuisances in each dataset: `Background`, `dark`, and `black` capture canvas/intensity, and for DecoyMNIST `corner`, `patch`, `box`, `corner patch` explicitly describe the spurious square so WeCLIP+ separates it from the foreground.

# C Hyperparameter Search




(a) ColorMNIST: Optim Value across $(\lambda, E_{\text{attn}})$.

(b) DecoyMNIST: Optim Value across $(\lambda, E_{\text{attn}})$.

Figure 2: **Hyperparameter heatmaps.** Each cell shows the best *Optim Value* during training (higher is better).

We perform a grid search over the KL weight $\lambda$ and the Attention epoch $E_{\text{attn}}$ (the epoch at which training switches from pure attention alignment to the combined objective in Eqn. 3). Each cell reports the *Optim Value* defined as

$$\text{Optim Value} = \text{ValAcc} \times \big(1 - \mathcal{L}_{\text{attn}}\big),$$

so *higher is better*. Training settings match the main text (SGD, 64 batch, 30 epochs); the initial learning rate is $10^{-3}$ for ColorMNIST and $10^{-2}$ for DecoyMNIST. $\mathcal{L}$ can be larger than 1, having a val optim number close to zero might still get decent accuracy on the test set, this is just a metric for optimizing hyperparameters

**Note on the selection metric.** Because $\mathcal{L}_{\text{attn}}$ is a KL divergence, it is non-negative and *unbounded*; in particular, $\mathcal{L}_{\text{attn}} > 1$ can occur. Consequently $1 - \mathcal{L}_{\text{attn}}$ may be small or even negative, so an *Optim Value* near zero does not imply poor validation or test accuracy. We use this quantity solely to *rank* hyperparameter settings during the grid search.

**Selected hyperparameters.** From the grid search in Fig. 2, the best settings we use in the main results are:

Table 2: Chosen $(\lambda, E_{\text{attn}})$ from the hyperparameter search. Higher Optim Value is better.

| Dataset | $\lambda$ | $E_{\text{attn}}$ | Optim Value |
|---|---|---|---|
| DecoyMNIST | 8 | 13 | 0.1015 |
| ColorMNIST | 160 | 11 | 0.1069 |