

A Details on AION-1 Models

We provide in Table 5 a description of the different configurations used for our suite of models.

Table 5: **Model size variants:** Following the choice of 4M model, we adopt the conventional T5 model sizes and naming schemes [47].

Model	Encoder Blocks	Decoder Blocks	Model Dim	Num Heads	Total Params
AION-1-B	12	12	768	12	300M
AION-1-L	24	24	1024	16	800M
AION-1-XL	24	24	2048	32	3B

Pretraining Details. We adopt an input budget of 256 tokens, and output budget of 128 tokens for all our models during pretraining. All models are trained with bfloat16 mixed precision, and model distribution under PyTorch’s Fully Sharded Data Parallel (FSDP) ZeRO-2 strategy. To achieve a batch size of 8192 in all cases, we train AION-1-B using 64 H100 GPUs for 1.5 days, AION-1-L using 100 H100 GPUs for 2.5 days, and 288 H100 GPUs for 3.5 days.

B Details on Tokenizers

B.1 Multi-Survey Image Tokenizer

We present a single image tokenizer capable of handling heterogeneous, multi-band images from different survey pipelines. Our approach unifies data from the Hyper Suprime-Camera (HSC) and the Legacy Imaging Survey (SGC)—surveys that differ in photometric filters, pixel scales, and noise characteristics—by normalizing each image to a common flux reference, merging all bands into a fixed-channel representation, and applying a subsampled linear embedding to account for variable channel counts. We then use a ResNet-based encoder–decoder with finite scale quantization (FSQ) to tokenize and reconstruct the inputs, training the model under an inverse-variance-weighted negative log-likelihood objective that leverages survey-specific noise estimates. Below, we detail the data sources, preprocessing, batching strategy, channel embedding, quantization settings, and loss function.

B.1.1 Data Sources and Band Information

We tokenize images from the Hyper Suprime-Camera (HSC) [4], covering both the HSC Deep and HSC Wide fields, and from the southern galactic cap (SGC) of the Legacy Imaging Survey [13] using the cuts and data preparation described in [63]. HSC provides images captured in five optical/near-infrared filters $\{g, r, i, z, y\}$ and Legacy Survey (SGC) provides images captured in four $\{g, r, i, z\}$. Not only do the total band counts differ between surveys, but each band covers slightly different central wavelengths and possesses different pixel scales and noise properties. For example, HSC g -band is centered at ~ 477 nm with a $\sim 0.168''$ pixel scale, while the Legacy Survey g -band is centered at ~ 481 nm with a $\sim 0.262''$ pixel scale. The native brightness units in both surveys are also recorded at different zero-point magnitudes¹: 22.5 nanomaggies for Legacy Survey and 27.0 nanomaggies for HSC. Finally, due to deeper exposures and finer pixel sampling, HSC typically exhibits lower per-pixel noise levels than the Legacy Survey. Due to these differences in band information, it is critical to construct a tokenizer which integrates provenance information for each channel (e.g. which telescope the data originates from) in the available image.

B.1.2 Constructing Mixed Batches

To train a single model to handle images from multiple surveys, we construct *mixed* batches, each containing images from distinct surveys. While it is possible to train on survey-specific batches separately, mixing them promotes consistent multi-domain generalization and allows every forward pass to incorporate diverse images. The steps for creating these batches are detailed below.

¹The zero-point magnitude defines the reference flux level corresponding to a magnitude of zero in a given photometric system.

750 **Normalization.** We first re-normalize each image to a standardized flux scale. Specifically, we
 751 convert each survey’s zero-point magnitude ZP into a multiplicative scale

$$s = 10^{(ZP-22.5)/2.5}, \quad (5)$$

752 then divide the image flux by s . Next, we multiply by the ratio of the pixel scale to that of the Legacy
 753 Survey. This ensures that each image’s flux is matched to a common reference (the Legacy Survey
 754 convention), ensuring that all images in each batch are normalized to the same scale regardless of
 755 survey.

756 **Channel Padding and Masking.** We construct heterogeneous batches by stacking images from
 757 multiple surveys. First, we unify all distinct bands into a single *fixed* set of 9 channels (5 from
 758 HSC and 4 from Legacy Survey), assigning a specific index to each channel. Next, we map every
 759 image into a 9-channel tensor, filling the subset of channels corresponding to that image’s bands
 760 with flux values and setting any unused channels to zero. Crucially, we maintain a corresponding
 761 *channel-mask* tensor to indicate which channels are valid versus zeroed-out. This mask prevents
 762 downstream components from conflating dummy (all-zero) channels with genuine data.

763 **Random Sampling.** To ensure that our tokenizer model regularly encounters images from both HSC
 764 and the Legacy Survey (SGC), we construct each mini-batch by randomly drawing a specified fraction
 765 of examples from each dataset. We use a sampling ration of 1:2:20 HSC Deep:HSC Wide:Legacy
 766 Survey SGC, and reset the iterator for exhausted surveys until all surveys have been exhausted.

767 B.1.3 Tokenizer Details

768 **Subsampled Linear Channel Embedding.** Before passing the multichannel image into the ResNet-
 769 based tokenizer, we apply a *subsampled linear projection* adapted from [38]. Let $\mathbf{x} \in \mathbb{R}^{B \times C \times H \times W}$
 770 denote a batch of images, where B is the batch size, C is the total number of possible channels (in
 771 this case, 9), and H, W are spatial dimensions. The corresponding channel mask is $\mathbf{m} \in \{0, 1\}^{B \times C}$.
 772 We first zero out any invalid channels:

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \mathbf{m}, \quad (6)$$

773 where the mask \mathbf{m} is broadcast in the spatial dimensions. Next, a linear transformation maps $\tilde{\mathbf{x}}$ into a
 774 higher-dimensional space of size dim_{out} . Since some channels are masked, we also normalize by the
 775 number of valid channels present in each sample:

$$\hat{\mathbf{x}} = \alpha(\mathbf{m}) \times (\tilde{\mathbf{x}} W + b), \quad (7)$$

776 with learnable parameters $W \in \mathbb{R}^{C \times \text{dim}_{\text{out}}}$, $b \in \mathbb{R}^{\text{dim}_{\text{out}}}$, and a scale factor $\alpha(\mathbf{m})$ designed so that
 777 the expected norm of $\hat{\mathbf{x}}$ does not degrade if fewer or more channels are available. In practice, dim_{out}
 778 is set to be roughly $6 \times$ the number of input channels (e.g., 80 if $C = 9$), which we empirically find
 779 yields the best downstream performance.

780 **Encoding and Decoding Model.** We feed $\hat{\mathbf{x}}$ from Equation B.1.3 into our ResNet-based tokenizer,
 781 adapted from [71]. The tokenizer encoder consists of 2 downsampling blocks, which reduce the
 782 dimensionality of the input image by a factor of 16. This is inverted during upsampling in the decoder.
 783 In total, the ResNet-based tokenizer has roughly 50M parameters.

784 At the bottleneck of the encoder, we implement the Finite Scale Quantizer [40] to tokenize our images;
 785 we detail the quantization below. After passing through the tokenizer’s encoder–decoder pathway, a
 786 decoded feature map, $\mathbf{z} \in \mathbb{R}^{B \times \text{dim}_{\text{out}} \times H \times W}$ is produced, where dim_{out} is the same dimension used
 787 during the forward subsampled projection (e.g., 54). We then invert the subsampled embedding listed
 788 above to produce a decoded sample.

789 **Quantization and Tokenization.** At the bottleneck of our ResNet-based tokenizer, we quantize
 790 features into a discrete set of codes. We experiment with multiple approaches, but empirically, we
 791 find that FSQ [40] yields the best performance in terms of reconstruction fidelity and training stability.
 792 Further, to explore the trade-off between reconstruction loss and codebook utilization, we vary the
 793 codebook size from smaller (e.g., 2^4) to larger (e.g., 2^{14}) and observe that a size of 2^{12} offers a
 794 desirable balance: the reconstruction loss plateaus with larger codebooks, while code usage remains
 795 sufficiently high to avoid underfitting with smaller codebooks; this is demonstrated in Figure 6.
 796 Consequently, our final configuration employs FSQ with a codebook size of 2^{12} .

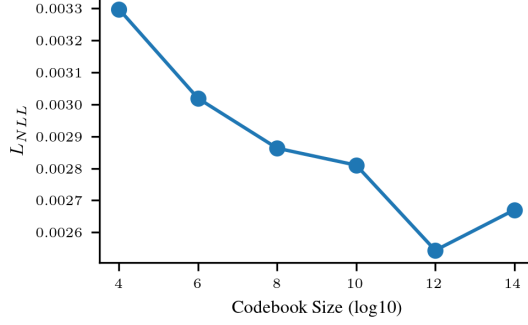


Figure 6: **Tokenizer Reconstruction Performance vs. FSQ Codebook Size.** We plot the MSE on a held-out validation set vs. the size of the FSQ quantizer used in the tokenization of the images. As shown, model performance appears to plateau after a codebook size of 2^{12} .

797 B.1.4 Loss Function and Per-band Weighting

798 Our training objective extends the standard mean-squared error (MSE) reconstruction loss to an
 799 inverse-variance-weighted Gaussian negative log-likelihood (NLL). In particular, because each image
 800 has a different number of valid bands, we only sum over valid channels (as tracked by the mask), and
 801 we additionally normalize by the number of valid channels per sample so that multi-band images do
 802 not dominate the loss compared to fewer-band images. Concretely, if $\mathbf{x}_i^{(c)}$ denotes the flux in channel
 803 c of image i , and $\sigma_i^{(c)}$ is its noise standard deviation, we compute:

$$\mathcal{L}_{\text{NLL}} = \sum_{i=1}^B \frac{1}{N_i} \sum_{c \in \mathcal{C}_i} \frac{1}{2} \left(\frac{\mathbf{x}_i^{(c)} - \hat{\mathbf{x}}_i^{(c)}}{\sigma_i^{(c)}} \right)^2, \quad (8)$$

804 where $\hat{\mathbf{x}}_i^{(c)}$ is the reconstructed flux, \mathcal{C}_i is the set of valid channels for image i , and $N_i = |\mathcal{C}_i|$ is the
 805 number of valid channels in that image.

806 B.1.5 Training Details

807 To optimize the tokenizer, we train the model end-to-end using Adam [28] with a learning rate of
 808 5×10^{-4} and a batch size of 64 split across 4 GPUs (i.e., each GPU sees a mini-batch of size 64).
 809 We employ a cosine decay learning rate schedule over the course of training, with a linear warmup
 810 of the first 1000 steps to stabilize convergence. We train over the entire dataset, including HSC
 811 Deep, HSC Wide, and Legacy Survey (SGC), which collectively comprise roughly 120 million image
 812 patches. This results in approximately 500,000 gradient steps, during which each dataset is sampled
 813 in proportion to the specified ratio above. The training process takes roughly 5 days on 4 NVIDIA
 814 H100 GPUs.

815 B.2 Spectrum Tokenizer

816 Here we describe the design of a unified spectral tokenization model which is capable of joint
 817 processing of spectra from different surveys and spectrographs, as well as joint processing of different
 818 objects; previous analyses of spectra have typically worked with a single object type (e.g., [39]).
 819 Below we detail the data sources, model architecture, tokenization strategy, and training details.

820 B.2.1 Training data

821 We train the spectrum tokenizer on a mix of SDSS and DESI spectra from The Multimodal Universe
 822 [63]. The SDSS dataset contains spectra taken by both the SDSS spectrograph, which covers
 823 $3800 - 9200 \text{ \AA}$, and the the BOSS spectrograph, which covers $3650 - 10400 \text{ \AA}$. The spectrographs
 824 have a resolution of $R = \lambda/\Delta\lambda = 1500$ at 3800 \AA and $R = 2500$ at 9000 \AA . DESI EDR spectra span
 825 $3600 - 9824 \text{ \AA}$, with a resolution of $R = 2000$ at 3600 \AA and $R = 5500$ at 9800 \AA .

826 Both datasets contain spectra with amplitudes that range orders of magnitudes, and both contain
 827 spectra of galaxies, stars, and quasars. The spectra for all of these types of objects display spectral

828 lines which are important indicators of physical properties. Small differences in the amplitude, width,
829 or position of any line can have a dramatic impact on the inferred properties. Thus, any tokenizer that
830 we design needs to have sufficiently high fidelity to capture the precise shapes and diversity of these
831 lines.

832 **B.2.2 Model Architecture and Tokenization Process**

833 Any input spectrum contains two channels: one for the flux, and one for the “inverse variance”, which
834 is the per-pixel uncertainty on the flux. The inverse variance is set to zero (i.e., infinite error) for bad
835 pixels, such as ones for which the sky/atmosphere is extremely bright relative to the object being
836 measured. Each spectrum is also associated with an **observed** wavelength grid.

837 To tokenize a spectrum, we first perform several preprocessing steps. We first calculate the median flux
838 of the spectrum, ignoring bad pixels. The median, which can vary significantly, is then compressed
839 with a log-transform to shrink the range of possible values, and then tokenized separately with a
840 scalar linear quantizer. The median is also used to normalize both the flux and the inverse variance
841 before they are stacked together channel-wise.

842 This stacked spectrum is then interpolated and resampled onto a latent wavelength grid, defined a
843 priori over a certain range. By using a latent wavelength grid and projecting input spectra to that grid,
844 we are able to handle spectra that come from different surveys, each of which have spectrographs that
845 measure at different wavelength ranges and resolutions. We then pass the stacked, normalized, and
846 projected input to a convolutional encoder based on the ConvNeXt V2 architecture.

847 The encoder consists of an initial downsampling stack composed of a 4x4 convolution and LayerNorm,
848 followed by three downsampling stacks of 2x2 convolutions and LayerNorms. Each of the four
849 downsampling stacks is followed by multiple ConvNeXt V2 processing blocks. The encoder ends
850 with a final LayerNorm at the end. Look-up Free Quantization is then applied to turn the processed
851 spectrum into tokens.

852 The decoder has a similar architecture to the encoder, but inverted (i.e., downsampling convolutional
853 layers are replaced with upsampling transposed convolutional layers). First, input tokens are dequan-
854 tized, then repeatedly upsampled and processed by ConvNeXt blocks. The final output has length
855 equal to the latent spectral grid and two channels. One channel is dedicated to flux and the other to a
856 “mask”, whose purpose is to indicate the pixels for which the model has made unreliable predictions.

857 **B.2.3 Loss Functions**

858 The output flux and mask are both projected onto a given observed wavelength grid to match a
859 particular survey. The flux is then denormalized, and a sigmoid function is applied to the output
860 mask.

861 The tokenizer is trained with three losses; there is one for each output channel, and one for the
862 quantizer. First, we use a Gaussian likelihood loss between the input and output fluxes, with the
863 weights given by the inverse variance, so that the output flux matches the input flux, with pixels that
864 are more certain being given more importance. Second, we use a binary cross-entropy loss between
865 the output mask and the input mask, so that mask resulting from the tokenizer gives an “unreliability
866 probability”, indicating the likelihood of a value being unreliable due to either a bad pixel or because
867 the model has not seen any input in that wavelength range. Finally, we have a quantization loss on
868 the LFQ module.

869 **B.2.4 Model Configuration and Training Details**

870 For the spectral tokenizer in AION-1, we use a latent spectral grid ranging from 3500 Å to 10462.4 Å
871 with a resolution of 0.8 Å (i.e., 8704 pixels), covering the wavelength range observed by the majority
872 of optical spectrographs. The number of ConvNeXt processing blocks in the various layers of the
873 encoder, in order, is 3, 3, 27, and 3, and the number of processing blocks in the layers of the decoder
874 is 3, 3, 9, and 3. The hidden dimensions in the encoder are 64, 128, 256, and 512; this is inverted
875 for the decoder. We use an LFQ module with an embedding dimension of 10, corresponding to a
876 codebook size of 1024. The normalization token is quantized separately using a scalar tokenizer,
877 again with a codebook size of 1024. For both the encoder and decoder, convolutional layer weights

are initialized with a truncated normal distribution with mean of 0 and a std of 0.02, and their biases are initialized to 0.

We train the tokenizer with the AdamW optimizer with $\beta_0 = 0.9$, $\beta_1 = 0.999$, and weight decay = 0.01, in bfloat16 mixed precision with a constant learning rate of 10^{-4} and a global batch size of 128. The commitment loss weight on the quantization loss is 0.25. We perform 215,000 gradient update steps, which takes 1 day on 4 NVIDIA H100 GPUs.

C Details on Galaxy and Stellar Parameter Inference

Data (Galaxy Parameter Inference). For inferring galaxy parameters, we use the data from the PRObabilistic Value-Added Bright Galaxy Survey (PROVABGS) [22], which provides derived physical properties from galaxy photometry and spectroscopy using a Bayesian Spectral Energy Density (SED) physical model. In particular, we extract:

- z : Redshift of the galaxy
- Z_{met} : Metallicity
- M_* : Stellar mass
- t_{age} : Stellar population age
- SFR: Star formation rate

We cross-match these galaxies with the Legacy Survey [13] imaging and photometry in the southern hemisphere and DESI spectroscopy [2]. Then, we apply several quality cuts to ensure reliable parameter estimates. In particular, we remove any objects with $M_* < 0$ as well as objects with any unphysical photometric magnitudes ($m < 0$). After these cuts, we arrive at a relatively clean sample of $\approx 100,000$ galaxies suitable for testing and validating our inference pipeline.

To accommodate the large dynamic range in certain parameters, we take the logarithm of both Z_{met} and M_* . We also convert SFR into the specific star formation rate (sSFR), given by

$$\text{sSFR} = \log \frac{\text{SFR}}{M_*}, \quad (9)$$

which is often more insightful for characterizing the relative growth of stellar mass in the galaxy.

Data (Stellar Parameter Inference). The *Gaia* dataset that we train AION-1 on is a subset of the available data in *Gaia* DR3. We cross-match this subset with DESI to produce the sample for which we evaluate stellar parameter inference. The training set and validation sets consist of the cross-matched stars belonging to the training and validation healpixes during pretraining to ensure that the validation is performed on stars that AION-1 has not seen during pretraining.

While the MMU dataset for DESI provides stellar spectra, we turn to the catalog of [73] (hereafter Z24) for stellar parameters. Z24 uses a data-driven method with regularization from physical models to provide estimates for basic stellar parameters like T_{eff} , $\log g$, and $[\text{Fe}/\text{H}]$, as well as for various abundances $[\text{X}/\text{Fe}]$ from DESI spectra.

Models. We perform parameter inference by training linear models on top of AION-1 embeddings. Given some data inputs, we first tokenize them, then extract the corresponding embeddings from the pretrained and frozen AION-1 encoder. We experiment with both mean-pooling and cross-attention to compress the sequence dimension of the embeddings before applying a linear projection layer to the channel dimension to produce estimates for each quantity of interest.

Data Fusion. We train all downstream models on various input combinations as a demonstration of data fusion. In particular, for galaxies we train downstream models on galaxy photometry, galaxy photometry and imaging, and galaxy photometry, imaging and spectroscopy. For stars: we train downstream models on stellar photometry, stellar photometry and low resolution spectroscopy, and low resolution spectroscopy and parallax.

Baselines. In the following section we detail the dedicated, supervised baselines used for galaxy and stellar property prediction.

We train two baseline models for stellar property prediction:

- **ConvNeXt Regressor on Raw Spectra.** This model uses multiple stacks of alternating ConvNeXt [69] processing blocks and downsampling blocks—identical in architecture to the encoder used for AION-1 spectrum tokenization—followed by attention pooling and a final linear projection. It is trained directly on the raw, pixel-level stellar spectra and noise estimates.
- **XGBoost on Token Representations.** We train an XGBoost regressor on the mean-pooled tokens produced by the AION-1 spectrum tokenizer. Because these tokens are already a compressed, high-level representation of the spectra, the XGBoost model’s task is, in principle, simpler than that of the ConvNeXt regressor, which must simultaneously learn to extract features and perform the final prediction.

For galaxy property prediction, we train three different baseline models:

- **XGBoost on Photometry.** We train an XGBoost regressor directly on the photometric measurements of galaxies (e.g., magnitudes or fluxes in various bands).
- **ConvNeXt-Tiny on Images.** As a second baseline, we adopt the ConvNeXt-tiny architecture trained on galaxy images. This approach matches the baseline provided in the *Multi-Modal Universe* (MMU) framework.
- **Convolutional + Attention Network on Spectra.** Inspired by the method of [39] and used as a baseline in [46], we train a network that combines convolutional layers with attention-based pooling on the galaxy spectra.

Performance and Scaling. We present below an extended set of results on stellar and galaxy parameter inference.

In Figure 8 we show the performance of AION-1-B, AION-1-L, and AION-1-XL, as well of the two baselines, on the prediction of individual properties as a function of training set size. We find that in general, AION-1 outperforms the baselines across the board, with the XGBoost baseline typically being slightly worse and the ConvNet baseline being significantly worse, especially in the low-data regime where only 100 training samples are available. However, while the ConvNet baseline performs very poorly with few samples, its performance improves with training set size up until the full training sample of $\sim 50,000$ examples, at which point it performs essentially identically to—and in some cases better than—AION-1-B/L/XL and the XGB baseline. It is possible that the regression task is not particularly difficult, and thus model performance saturates early; with AION-1-B performing essentially the same as AION-1-L, we find that scaling up to a large model provides little benefit **for this task**.

Figure 7 shows that overall, independent of input modalities, cross-attention pooling (solid lines) significantly outperforms mean-pooling (dashed lines).

D Details on Dense Predictions

In this section, we provide additional details on our dense fine-tuning experiments, which involve predicting segmentation maps and detecting sets of objects from image inputs. While prior work by [42, 6] classifies object detection as a sparse prediction task—treating it as an autoregressive sequence generation problem—we refer to it as dense prediction in contrast to our scalar prediction tasks. This distinction emphasizes the structured nature of segmentation and object detection compared to simpler regression-based outputs.

D.1 Architecture

Semantic Segmentation. We implement a lightweight convolutional upsampler trained on top of AION-1’s encoder representations. Our upsampler design is largely inspired by the mask decoder

	z		M _*		t _{age}		Z _{Met}		SFR	
	Mean	Attention	Mean	Attention	Mean	Attention	Mean	Attention	Mean	Attention
AION-1-B										
<i>Ph</i>	0.736	0.754	0.714	0.720	0.350	0.353	0.409	0.412	0.410	0.378
<i>Ph+Im</i>	0.910	0.934	0.857	0.886	0.394	0.445	0.453	0.490	0.611	0.637
<i>Ph+Im+Sp</i>	0.779	0.995	0.739	0.956	0.258	0.532	0.365	0.610	0.439	0.720
AION-1-L										
<i>Ph</i>	0.659	0.761	0.630	0.734	0.268	0.357	0.302	0.411	0.228	0.387
<i>Ph+Im</i>	0.922	0.940	0.870	0.889	0.412	0.454	0.460	0.496	0.621	0.642
<i>Ph+Im+Sp</i>	0.800	0.995	0.760	0.955	0.276	0.534	0.375	0.620	0.461	0.727
AION-1-XL										
<i>Ph</i>	0.679	0.792	0.647	0.757	0.266	0.314	0.318	0.379	0.240	0.475
<i>Ph+Im</i>	0.910	0.940	0.857	0.888	0.394	0.450	0.439	0.490	0.610	0.644
<i>Ph+Im+Sp</i>	0.795	0.992	0.759	0.947	0.273	0.534	0.374	0.621	0.454	0.731
	Supervised		Supervised		Supervised		Supervised		Supervised	
<i>Ph</i> ¹	0.708		0.692		0.301		0.301		0.377	
<i>Im</i> ²	0.864		0.821		0.445		0.489		0.638	
<i>Sp</i> ³	0.998		0.852		0.433		0.621		0.675	

Table 6: **R² (↑) for galaxy property estimation.** Inputs to the model are: photometry (*Ph*), photometry and imaging (*Ph+Im*), and photometry, imaging, and spectra (*Ph+Im+Sp*). Mean implies taking the average over AION-1 embeddings followed by a linear projection head, while Attention implies training a cross-attention layer with a linear projection head on the full set of embeddings. Supervised models are: ¹XGBoost, ²ConvNext, ³Convolution + Attention Network. All models are trained on $\sim 100,000$ examples.

from [29]², but with a key modification: we do not include hypernetworks instantiated from additional register tokens. Instead, we use a single convolutional layer to project the upsampled output to the desired number of segmentation maps, simplifying the architecture while maintaining efficiency.

Clump Detection. For clump detection, we introduce no additional model parameters. Instead, we finetune AION-1’s decoder to autoregressively generate linearized object tokens, following the same tokenization scheme used in our pre-training catalog data. This approach enables the model to predict structured object sequences without requiring task-specific modifications.

D.2 Galaxy Zoo 3D Segmentation

Galaxy Zoo 3D is a dataset derived from volunteer annotations of galaxies, originally presented in [37] and collected through the Zooniverse³ citizen science platform. Each galaxy sample was annotated by 15 volunteers, who were asked to mark the galactic center, any stars in the frame, and to draw bounding boxes around galactic bars and spiral arms.

For this study, we focus on the vote maps, which consist of four dense arrays containing pixel-wise annotation counts (ranging from 0 to 15) indicating how many annotators included a given pixel in their annotations. Segmenting bars and spiral arms is particularly challenging, which is why human annotations are crucial for this task.

Following [66], we filter out samples that do not reach a confidence level of 0.2 and compare our results to a model proposed in the same study, which we trained ourselves. The ground truth segmentation is defined as the set of pixels that received any number of votes from annotators, normalized by the maximum number of votes received per sample.

For our dense predictions, we finetune a small convolutional head on top of AION’s frozen encoding module and optimize the trainable parameters using mean squared error (MSE). Figure 11 provides example vote maps, and we present our results in Table 3.

²<https://github.com/facebookresearch/segment-anything/>

³<https://www.zooniverse.org/>

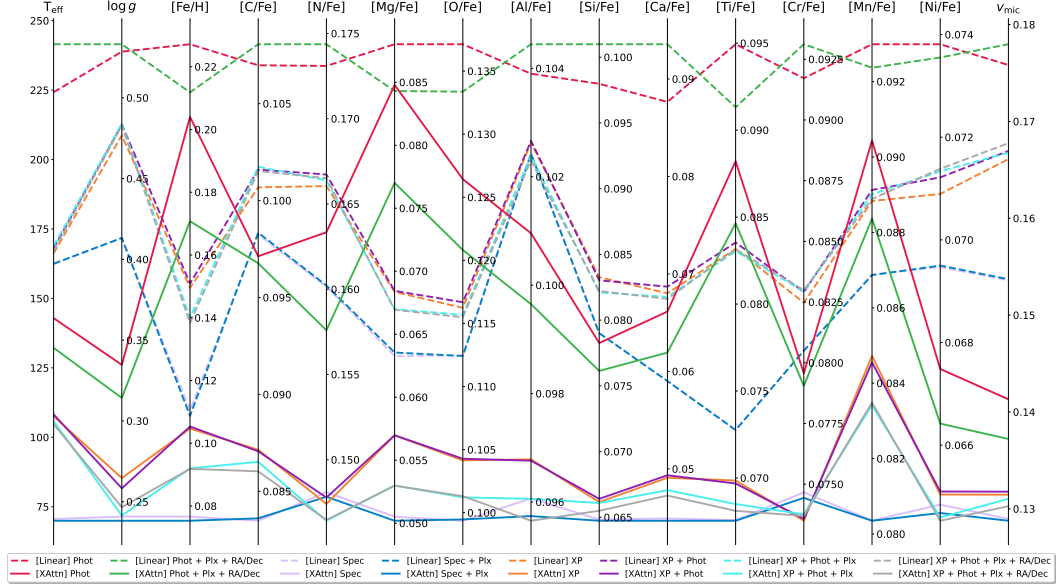


Figure 7: Predictive performance as measured by standard deviation of residuals on a held-out evaluation set of models with various input combinations on various physical properties of stars. **Lower is better.** Each line represents a different model, with dashed lines indicating mean-pooled linear probes and solid lines indicating cross-attention pooled linear probes. The color of the line represents the inputs that the model is given. All embeddings are generated from the frozen, pretrained encoder of Aion-1-L.

We evaluate our model using the Intersection over Union (IoU) metric, reporting separate IoU scores for spiral arms and bars rather than a combined mean IoU (mIoU). The IoU is defined as:

$$\text{IoU} = \frac{|\hat{\mathbf{M}} \cap \mathbf{M}|}{|\hat{\mathbf{M}} \cup \mathbf{M}|}, \quad (10)$$

where $\hat{\mathbf{M}}, \mathbf{M} \in [0, 1]^{H \times W}$ denote the predicted and ground truth segmentation masks, respectively.

We determine separate segmentation thresholds for spiral arms and bars using 20% of our validation set. These thresholds are computed independently, as annotators tend to agree more consistently on bar structures than on spiral arms. This approach ensures optimal segmentation performance for both components.

D.3 Galaxy Zoo Clump Detection

We additionally investigate the Seq2Seq problem of autoregressively generating galaxy clumps [14]. Specifically, we finetune AION-1 to generate an ordered sequence of clumps – where the number of clumps varies across different examples – by conditioning on Legacy Survey Images. To achieve this, we use the catalog tokenizer used during pre-training, which encodes catalog objects into a structured sequence of quintuples, each consisting of pixel coordinates, elliptical shapes, and radius. We then finetune the model with a causal language modeling objective, conditioning on both the previously generated clumps and the corresponding Legacy Survey images. This setup allows the model to learn spatial and morphological dependencies among clumps, ultimately improving its ability to generate realistic clump distributions for galaxy images. This dataset consists of 3727 cross-matched samples. Some qualitative examples are shown in Figure 10.

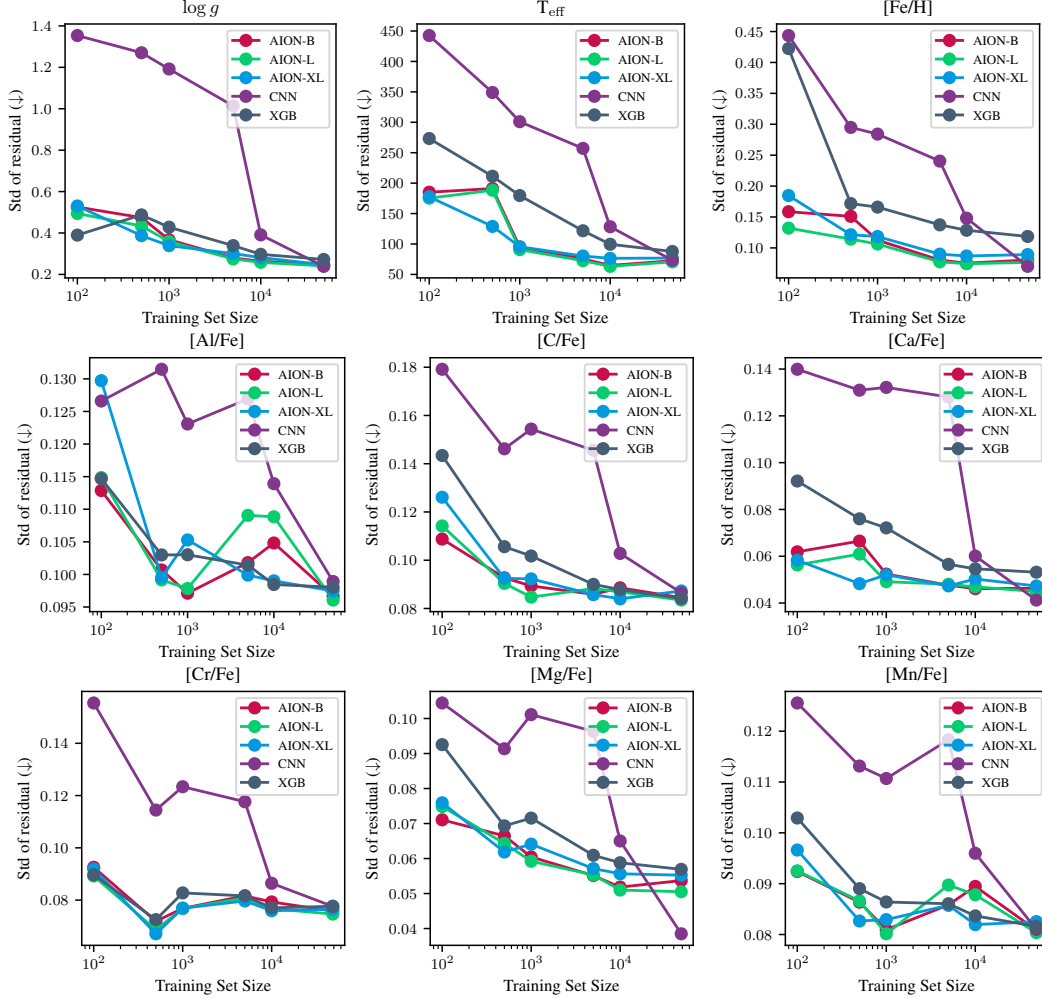


Figure 8: Predictive performance of AION-1-B/L/XL, a convolutional baseline from raw spectra, and an XGBoost baseline from spectrum tokens as a function of training set size for different stellar properties. Performance is measured in terms of the standard deviation of the residuals, and thus **lower is better**.

1009 E Details on Retrieval Evaluation

1010 E.1 Evaluation Metric

1011 For each query galaxy, we first produce an embedding using AION-1. We then generate embeddings
 1012 for all other galaxies in the search corpus. All embeddings are averaged together to produce a single
 1013 vector, $\mathbf{x} \in \mathbb{R}^d$, where d is the embedding dimension of the specific AION model used. Next, we
 1014 compute the cosine similarity between the query embedding, \mathbf{x}_q and each candidate embedding \mathbf{x}_c ,
 1015 as

$$S_c(\mathbf{x}_q, \mathbf{x}_c) = \frac{(\mathbf{x}_q)^T \mathbf{x}_c}{\|\mathbf{x}_q\|_2 \|\mathbf{x}_c\|_2}. \quad (11)$$

1016 The entire corpus (excluding the embedding) is then ranked in descending order of similarity. We
 1017 compute the normalized Discounted Cumulative Gain (nDCG) retrieved objects, where the relevance,
 1018 r_i of each object, i , is determined by the criteria described in the sections below. Specifically, the
 1019 DCG at rank k is defined as:

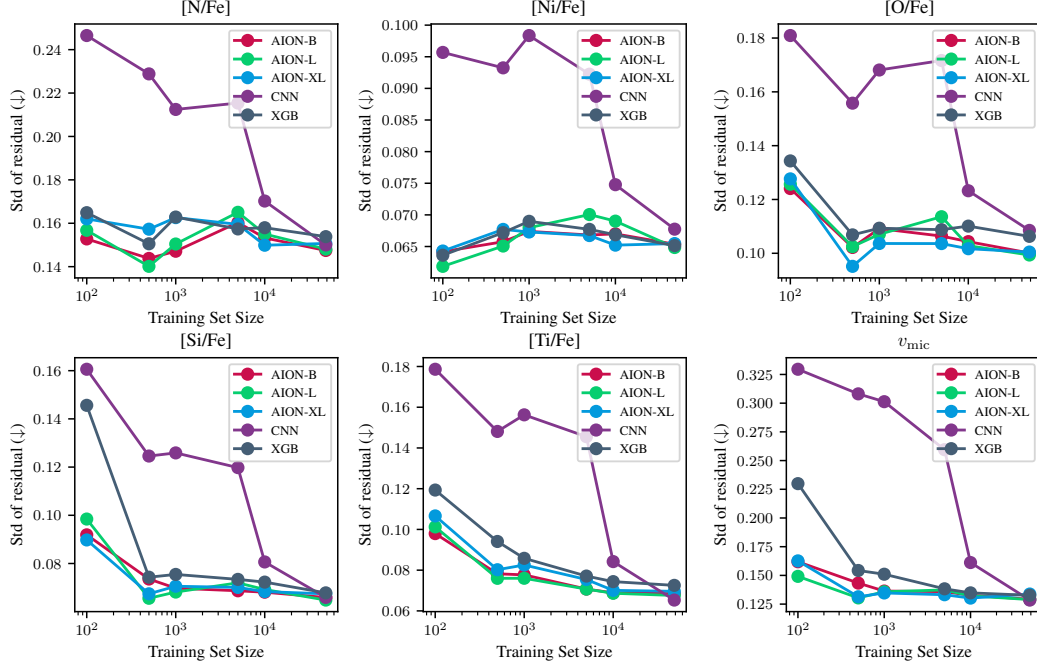


Figure 8: (continued)

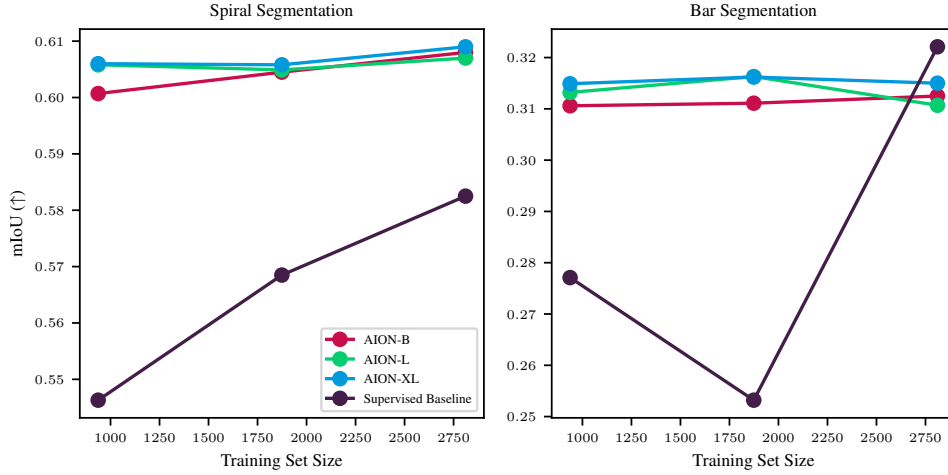


Figure 9: **IoU scores for spiral arm and bar segmentation** across different models, evaluated at three fractions of the available training data 33% (937), 66% (1874), and 100% (2811).

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad (12)$$

1020 The ideal DCG (IDCG) is computed by sorting the items by descending relevance. The normalized
1021 DCG is then

$$\text{nDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}. \quad (13)$$

1022 In our experiments, we focus on $k = 10$ and report nDCG@10 as our evaluation metric.

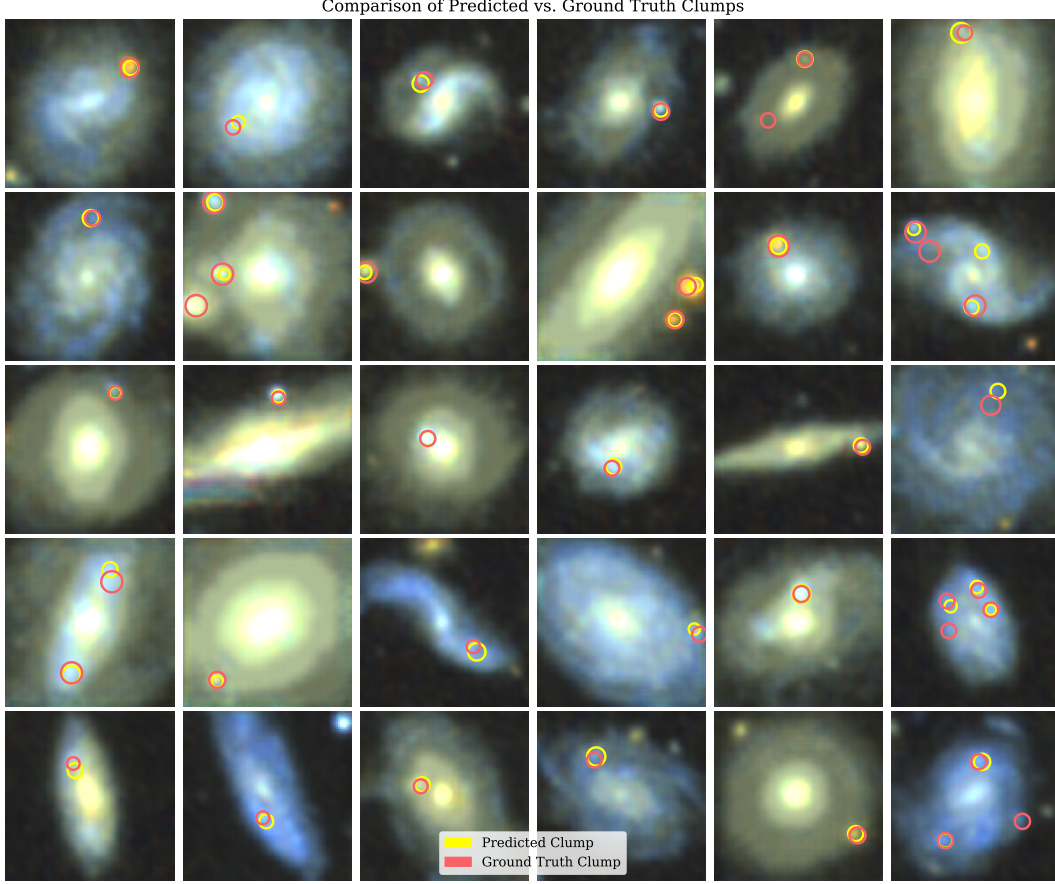


Figure 10: Qualitative examples of ground truth and predicted clump objects in Legacy Survey.

1023 E.2 Galaxy Zoo Object Labels

1024 **Data.** We use the Galaxy Zoo DECaLS catalog [65], which provides citizen-science morphological
 1025 classifications for a large set of galaxies observed in the Legacy Imaging Survey [13]. We then
 1026 cross-match these classified objects with corresponding Legacy Survey South images. To ensure that
 1027 each galaxy has sufficiently reliable volunteer annotations, we discard any objects with fewer than
 1028 three total volunteer votes. This leaves $\approx 171,000$ galaxies.

1029 Within this filtered dataset, we focus on two morphological classes of interest: *mergers* and *spirals*.
 1030 We select high-confidence examples of these two classes by identifying galaxies for which more than
 1031 $f = 0.9$ of the volunteers have voted for the corresponding morphology (merger or spiral). These
 1032 high-confidence galaxies form our set of *query objects* in the retrieval experiments. In total, we have
 1033 726 merging galaxies and 24,622 spiral galaxies.

1034 For each object in the dataset, we define a *relevance label* based on the fraction f of volunteers who
 1035 voted for the same morphological class. Thus, an object whose volunteer vote distribution aligns
 1036 more strongly with merger or spiral morphology is assigned a higher relevance label than one whose
 1037 distribution is more ambiguous. This setup encourages the retrieval model to prioritize both the
 1038 correct morphological class and the degree of confidence in that classification.

1039 **Aggregated Query Results.** In addition to evaluating single query objects, we investigate whether
 1040 *aggregating* multiple queries can improve retrieval performance. Specifically, instead of using a single
 1041 galaxy embedding as the query, we compute a single query embedding by *averaging* the embeddings
 1042 of multiple galaxies from the same morphological class (merger or spiral). The rationale is that
 1043 features indicative of the given morphology will be reinforced across several galaxy embeddings,
 1044 while idiosyncratic features unrelated to that morphology will be muted.

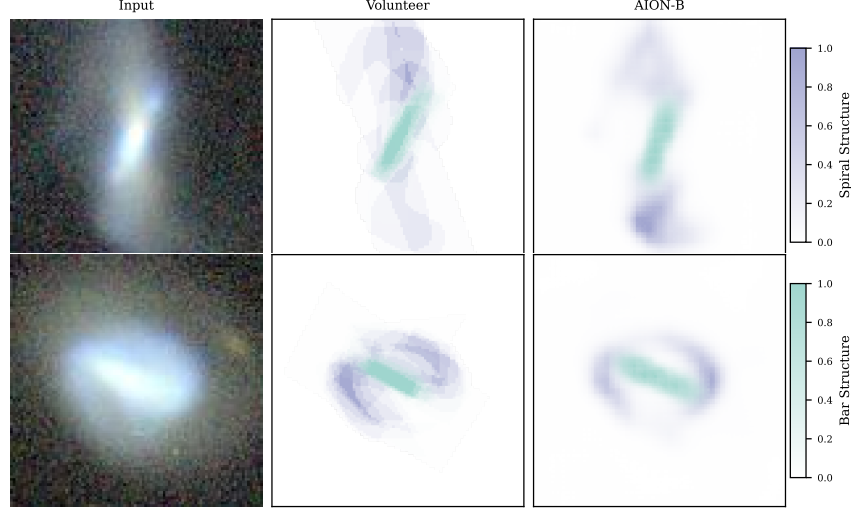


Figure 11: Exemplar ground truth image, crowd-annotated segmentation map, and AION-1 predictions.

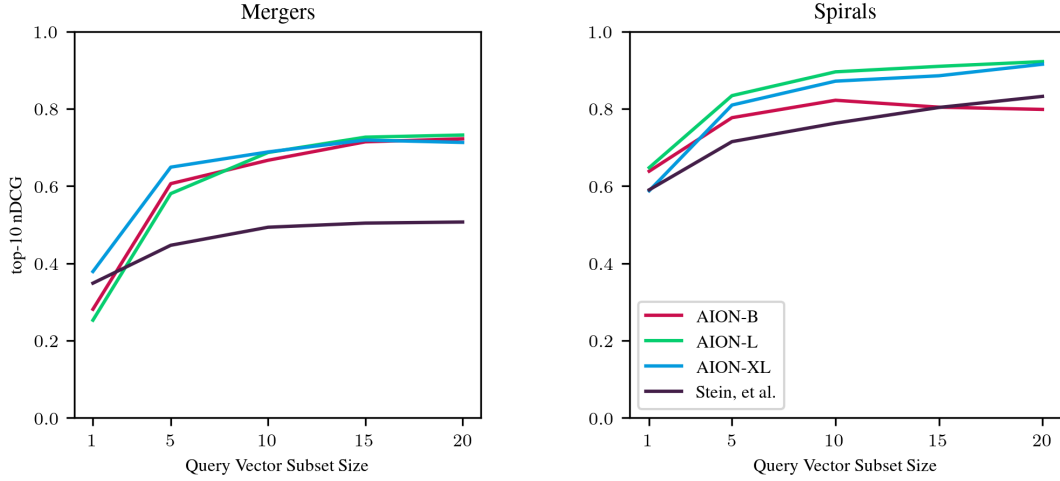


Figure 12: **nDCG@10 as a Function of Aggregated Query Size.** Overall, we find that mean-pooling multiple queries of the same object type to form our query vector dramatically improves model performance when searching for mergers and spirals, up to a query count of roughly 15. This holds for all models.

Figure 12 shows how the nDCG@10 score varies with the number of query embeddings being averaged. We observe that performance systematically increases up to about 15 queries, after which gains plateau. In particular, for merger queries, this approach *more than doubles* the nDCG@10 compared to single-query retrieval, while for spirals it boosts nDCG@10 by approximately 0.25. Notably, even with aggregated queries, our best models still outperform the baseline method of [60], underscoring the effectiveness of this aggregation strategy.

E.3 Strong Lens Finding

Data. For the strong lensing retrieval task, we start by filtering the cross-matched catalog of objects within the Legacy Survey and HSC datasets to approximately reproduce the parent sample used in the HSC strong lensing searches [27]. Specifically, we impose three additional cuts: (1) objects with photometric redshifts between 0.2 and 1.2, (2) objects with an estimated stellar mass above $5 \times 10^{10} M_{\odot}$, and (3) objects with a star formation rate to stellar mass ratio less than 1×10^{-10} . In order to identify the strong gravitational lenses within the resulting parent sample, we cross-match

	LS \rightarrow LS	HSC \rightarrow HSC	LS \rightarrow HSC	HSC \rightarrow LS
AION-1-B	0.012	0.018	0.004	0.016
AION-1-L	0.011	0.019	0.004	0.017
AION-1-XL	0.015	0.015	0.004	0.012
[60]	0.007	–	–	–

Table 7: **nDCG@10 for Strong Lensing Retrieval.** We evaluate retrieval performance on strong gravitational lenses, measuring nDCG for galaxies retrieved via cosine similarity of AION-1’s average token embeddings. Each column is labeled $\mathbf{X} \rightarrow \mathbf{Y}$, \mathbf{X} being the modality used to produce the query embedding and \mathbf{Y} the key embedding. For all four columns, we show the results for the three AION-1 variants (B, L, XL). Since the state-of-the-art self-supervised baseline only generates legacy survey (LS) embeddings, we only show its results on the **LS \rightarrow LS** task.

with previous lens-finding catalogs [55, 68, 56, 26, 57, 67, 27, 10, 53, 51, 33, 61, 7, 9, 8, 49, 43, 16, 24, 21, 45]. Most strong lensing catalogs offer a grade for each candidate. Since the criteria for this grading varies between catalogs, we ignore these grades and instead assign a relevance score of 1.0 to all the strong lensing candidates found within each catalog. All other objects in our parent sample are given a relevance score of 0.0. Even with the additional filtering, strong gravitational lenses make up only 0.1% of our parent sample.

For each object in the parent sample, we have both the HSC and Legacy Survey observations. For AION-1, we extract the embeddings by passing the tokenized observation through the frozen encoder. We then average the output over the patch dimension. For the state-of-the-art baseline model [60], we directly use the representation output by the model. Unlike AION-1, the baseline is trained solely for legacy survey images. The resulting AION-1 embeddings have 768 dimensions for both modalities, whereas the baseline model embeddings have 128 dimensions for the Legacy Survey modality.

Results. Since we have both HSC and Legacy Survey (LS) embeddings for our strong lens catalog, we can perform two retrieval tasks within a modality: LS query with LS keys (**LS \rightarrow LS**) and HSC query with HSC keys (**HSC \rightarrow HSC**). We can also explore two retrieval tasks between modalities: LS query with HSC keys (**LS \rightarrow HSC**) and HSC query with LS keys (**HSC \rightarrow LS**). The state-of-the-art baseline only enables **LS \rightarrow LS**. The nDCG@10 metrics for these tasks are reported in Table 7.

We find that we outperform the state-of-the-art on the **LS \rightarrow LS** task for all three AION-1 model sizes, with the largest model leading to the greatest performance improvement. Switching from the **LS \rightarrow LS** task to the **HSC \rightarrow HSC** task leads to further gains in the retrieval metric, confirming that AION-1 is successfully leveraging the higher-resolution information present in the HSC images. For the cross-modality tasks we find mixed performance for AION-1. On the **LS \rightarrow HSC** task we get the worst retrieval performance of any modality combination, but on the **HSC \rightarrow LS** task we get equivalent retrieval performance to the **HSC \rightarrow HSC** task. One possible cause is that our retrieval depends on informative query embeddings. For example, in Appendix subsection E.2 we find that aggregating query embeddings leads to significant improvements in performance on galaxy morphology retrieval. For strong lensing retrieval, the HSC embeddings are derived from more informative (higher-resolution) observations than the LS embeddings. This leads to better performance when we query with the more informative HSC embeddings over the LS embeddings.

The low overall nDCG@10 score for all models and tasks reflects the inherent challenge in retrieving strong-lensing images. The Einstein ring that characterizes a strong gravitational lens is often dim compared to other features in the image, and strong lenses themselves are incredibly rare events. Despite this challenge, the state-of-the-art model we outperform has already been used to identify 1192 new strong lensing candidates [59].

F Details on Model Scaling

We evaluate how the model size impact its performances. The evaluation computes the categorical cross-entropy on predicted token outputs given input tokens that are randomly selected among every available modality. The overall evaluation loss is the weighted average of each modality loss. Figure 13 reports the overall evaluation loss and the image and spectrum losses for the legacy and

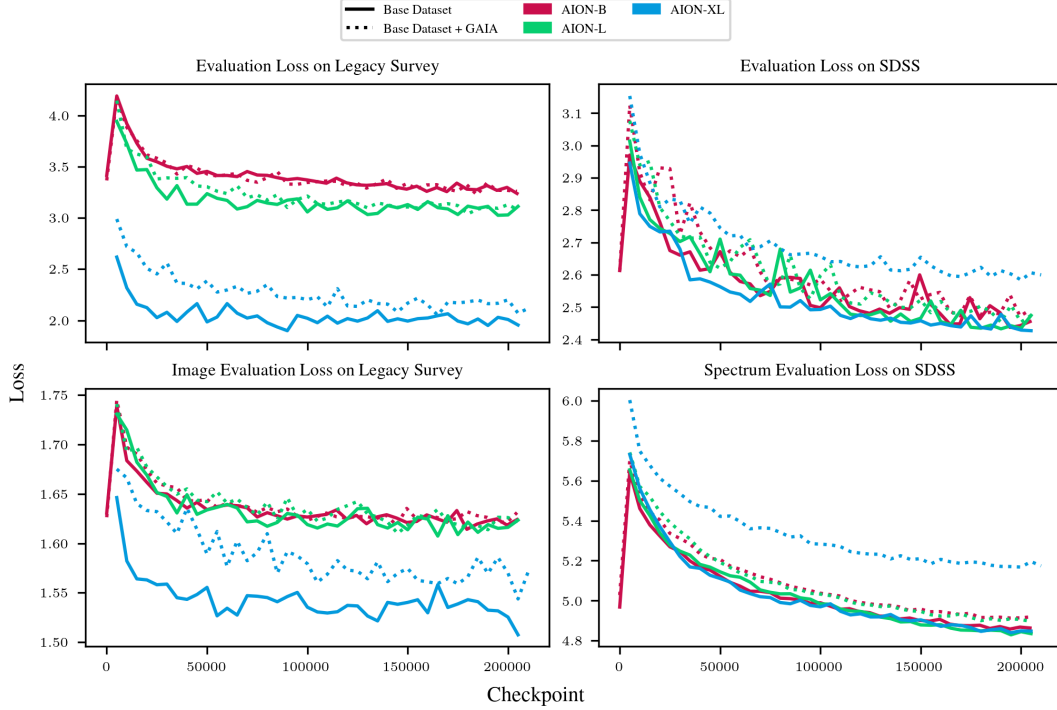


Figure 13: Evaluation losses for different model sizes on Legacy and SDSS surveys including image and spectrum modality. Base dataset refers to the training on all surveys except GAIA, while Base Dataset + GAIA are the models trained on all surveys described in Figure 2a.

SDSS surveys respectively. The decrease we see in the overall evaluation loss for the legacy survey indicates the model performs better when its size increases. However, the evaluation loss on SDSS remains similar regardless the size of the model. When checking for modality specifically (second row of Figure 13), it appears the image evaluation loss decreases with the size of the model while the spectrum evaluation loss stagnates. This indicates the amelioration observed while scaling the model is largely due to better performances in predicting token images. This trend could be explained by the fact that the complete dataset contains much more samples with images than with spectra (Figure 2a). There might not be enough spectrum data to observe improvement while scaling the model.

Additionally, we evaluate the impact of adding the GAIA dataset to the training set. GAIA contains 77M objects with spectrum data, thus should theoretically bring more information about this modality. Figure 13 compares the evaluation losses for different model sizes trained on all surveys except GAIA (Base Dataset i.e. Legacy survey + SDSS + DESI + HSW) and on all surveys including GAIA (Base Dataset + GAIA). For both evaluations on Legacy and SDSS surveys the loss is higher when training with Base Dataset + GAIA. Adding GAIA survey to the training dataset seems thus to impacts negatively the performances of the model on the other surveys. It is the case for image and even spectrum modality. The decreased performance on spectrum modality of SDSS survey, while adding spectrum information from GAIA in the training dataset, could be explained by the fact that GAIA spectrum is of much lower resolution than the one of SDSS.

G Full Modality Tokens

Category	Description	Modality
Imaging (2)	Legacy Survey imaging	tok_image
	HSC Wide imaging	tok_image_hsc
Catalog (1)	Legacy Survey catalog	catalog
Spectra (2)	SDSS spectra	tok_spectrum_sdss
	DESI spectra	tok_spectrum_desi
Gaia (4)	Gaia BP spectrum	tok_xp_bp
	Gaia RP spectrum	tok_xp_rp
	Gaia parallax	tok_parallax
	Sky coordinates	tok_ra, tok_dec
Gaia Photometry (3)	Gaia G-band flux	tok_flux_g_gaia
	Gaia BP-band flux	tok_flux_bp_gaia
	Gaia RP-band flux	tok_flux_rp_gaia
Legacy Survey (9)	g-band flux	tok_flux_g
	r-band flux	tok_flux_r
	i-band flux	tok_flux_i
	z-band flux	tok_flux_z
	WISE W1 flux	tok_flux_w1
	WISE W2 flux	tok_flux_w2
	WISE W3 flux	tok_flux_w3
	WISE W4 flux	tok_flux_w4
	E(B-V) extinction	tok_ebv
Legacy Survey Shape (3)	Ellipticity component 1	tok_shape_e1
	Ellipticity component 2	tok_shape_e2
	Effective radius	tok_shape_r
HSC Photometry (5)	g-band magnitude	tok_mag_g
	r-band magnitude	tok_mag_r
	i-band magnitude	tok_mag_i
	z-band magnitude	tok_mag_z
	y-band magnitude	tok_mag_y
HSC Extinction (5)	g-band extinction	tok_a_g
	r-band extinction	tok_a_r
	i-band extinction	tok_a_i
	z-band extinction	tok_a_z
	y-band extinction	tok_a_y
HSC Shape (3)	Shape component 11	tok_shape11
	Shape component 22	tok_shape22
	Shape component 12	tok_shape12
Other (1)	Redshift	tok_z

Table 8: Complete list of modalities used in our multi-survey analysis. The modalities are grouped by their source surveys and measurement types. Numbers in parentheses indicate the count of modalities in each category.