Rubric-Constrained Figure Skating Scoring

Arushi Rai
University of Pittsburgh
arr159@pitt.edu

Adriana Kovashka
University of Pittsburgh
kovashka@cs.pitt.edu

Abstract

Figure skating automatic scoring is the task of estimating the competition score of a performance video. The technical element score (TES) aggregates the technical quality (grade of execution) and difficulty (base value) scores for each element. Most prior work, adapted from short-term action quality assessment, entangle difficulty and quality, and compute TES for the entire video, reducing interpretability for athletes. This is mainly due to a lack of element segmentation and difficulty annotations in existing datasets. Motivated by increasing interpretability, we propose a novel method that implicitly segments a video to produce element-level representations and uses adherence with a natural language rubric to score each element, without needing additional annotations. We compute element-level representations using learnable element queries in a transformer and propose implicit segmentation regularization to encourage element queries to attend to elements rather than background transitions between elements (most of video). Additionally, we use the element list (sequence of elements) to isolate difficulty, just like judges who receive the routine list in advance, so we can focus on the more critical problem of how well elements are done. These components significantly improve interpretability, scoring precision, and ranking capability. Code is released at https://arushirai1.github.io/rcs-project.

1. Introduction

To the untrained eye, figure skating lacks the clear environmental feedback or visible outcomes that typically determine how points are scored in other sports (e.g., a ball going through a hoop). Instead, each performance is scored by judges who award points based on adherence to rubric items that evaluate quality, difficulty, technical proficiency, and artistic expression. While this scoring may seem subjective, judges are fairly consistent in their scores for a given performance; a panel of judges will agree on scores 96% of the time [14]. This consistency is achieved through detailed rubrics and scoring protocols.

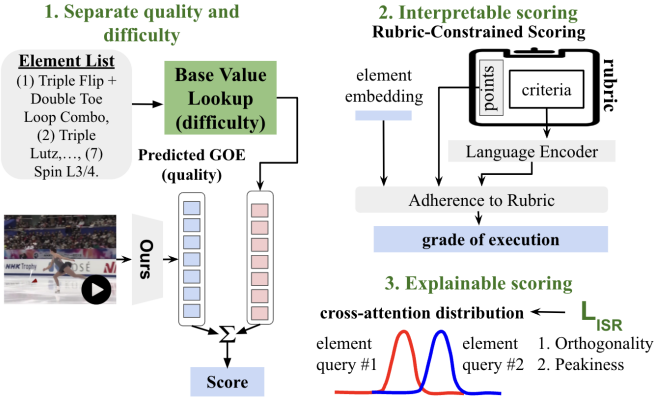

Figure 1. **Key concepts of our approach**: (1) The model focuses solely on predicting quality, not difficulty; (2) Scoring is interpretable, using rubric criteria to compute grade of execution from element embeddings; (3) Scoring is explainable, with element embeddings computed from cross-attention between the element query and clips. The cross-attention distribution is regularized by element segmentation priors to focus on relevant elements, not the background, without needing additional annotations.

This underlying structure behind subjective scoring in Olympic sports led to substantial interest in developing machine learning models for automatic action quality assessment. One sport whose automated scoring has been partly explored is figure skating. Athletes in competitive figure skating would benefit from automatic action quality assessment during personal practice if a coach or judge is not available. Such a system would need to be interpretable by the athlete to garner insights on how to improve their performance. However, scoring of long video activities, such as figure skating, is challenging for a number of reasons. We focus on two gaps in the literature.

First, long video activities like figure skating consist of a few elements (actions) over minutes of performance. Performance quality can vary throughout the video where some actions may be well done and others may be executed poorly. This calls for element-level scores, however, these long video activity datasets are sourced from publicly available sports broadcasts that do not have action segment an-

notations. Thus, action quality assessment methods learn to score the entire video and it is elusive how much each action contributed to the score, causing a challenge with interpretability. In contrast, during competitions, the scoring rubrics provide key insights into areas needing improvement and aspects that are done well.

Second, score prediction, as formulated in prior work, overfocuses on one component of scores, leading to mispredictions. Competition scores consists of two subscores: (1) holistic program component score (PSC); and (2) technical element score (TES) which aggregates (2a) technical quality (grade of execution, GOE) and (2b) difficulty (base value, BV), scored for each element. While TES combines both difficulty and quality, the variation in TES is often dominated by difficulty (base values) due to magnitude. Further, difficulty is easier to estimate since it only requires recognition whereas quality requires estimating how well multiple latent criteria are satisfied. Thus, existing figure skating AQA systems can get away with only estimating difficulty, rather than quality, providing limited value.

Our goal is to develop an interpretable scoring mechanism inspired by rubrics developed by the International Skating Union (ISU), which scores each element individually, and a single-stage architecture capable of providing element-specific scores without a separate segmentation model. Such a model should benefit athletes by offering accurate and understandable feedback without additional annotation burdens. While we choose to focus on figure skating, other sports also have rubrics and involve longer performances which would benefit from this type of architecture: gymnastics (artistic and rhythmic), synchronized swimming (artistic swimming), dressage (equestrian), freestyle skiing and snowboarding, ice dancing, martial arts, and cheerleading and dance competitions.

To utilize the element-level rubric items to score each element, we develop an architecture capable of scoring each element, rather than the whole video, and isolate learning quality only, rather than both difficulty and quality. To prevent scoring only difficulty, we follow a protocol that judges use: we use the performance's list of elements to look up the associated difficulty in advance, just like judges. We score elements in parallel using our proposed element transformer which produces element embeddings. We then use rubric text from the ISU scoring protocol documents, represented as text embeddings, to predict *grade of execution* (i.e., the term for the quality score in figure skating) for each element as shown in Fig. 1. We further improve the correspondence of the element embeddings to element segments through regularization. This regularization is inspired by the definition of a segment; it guides the queries in the decoder transformer layers to have cross-attention patterns where (1) each query attends to different parts of the video and (2) attention is concentrated on contiguous clips rather than scattered across the video. We call this implicit segmentation because the cross attention between the queries and clips should implicitly attend to element segments without requiring training over annotated segments. This approach aims to be interpretable through the use of rubric text for scoring and the long video architectural improvements help set up the use of rubric text and improve the accuracy of explanations and scoring performance.

Our proposed method outperforms state-of-the-art techniques and our strong baselines on the Fis-V benchmark [24] for figure skating technical element scoring in terms of both scoring precision (measured by mean-squared error) and ranking capability (measured by Spearman rank correlation). We also show results on FS1000 [21] in supp. Our method achieves significantly higher scoring precision, reducing the mean-squared error from 19.05 and 19.53 to 9.34 compared to the state-of-the-art and our best proposed baseline, respectively. To assess the impact on implicit segmentation, we annotated a small subset of the Fis-V test set with element segmentation (56 elements). We find a drastic increase in element queries attending to element segments in the video rather than background transitions (movements between elements) with our full method (12.5% to 35.7%). Our method also improves the Spearman rank correlation between our predicted TES scores and ground truth TES scores, increasing from 0.70 to 0.84 compared to the state-of-the-art. It achieves interpretability and performance without compromising on either.

2. Related Works

Action quality assessment. Automatic action quality assessment (AQA) was first explored by [14] using pose features and handcrafted spatiotemporal descriptors to assess Olympic diving and figure skating. [11, 22] showed improvements by fine-tuning spatiotemporal video-based architectures pretrained on large-scale action recognition datasets like Kinetics and UCF101. [12] introduced the AQA-7 dataset for short Olympic sports actions; while this is a common benchmark in AQA, our focus is on longer, structured activities so we evaluate only on figure skating. [24] found LSTM architectures more effective than average-pooling for longer videos like figure skating. [5] address long video skill assessment outside of sports which isn't applicable to our work on structured, rubric-based activity assessment. Some methods incorporate judging insights similar to our approach. [19] propose an uncertainty-aware method that predicts a score distribution to account for multiple judges and decouples difficulty from execution quality for single-action videos, while we focus on multi-action videos. [7, 25] use exemplars for relative scoring, which is unsuitable for longer activities like figure skating, where selecting the appropriate exemplar is challenging due to multiple sub-actions and varying sequences.

Use of language. [13] release a multi-task dataset with commentary for diving events, which is used to train on a captioning task in addition to scoring regression. [20] collect gesture error feedback from expert surgeons for surgical skill assessment but do not use these captions. [6] collect and uses commentary to distill knowledge from a teacher model. Our method leverages the semantic knowledge in CLIP text embeddings for scoring without relying on external language information per video.

Figure skating-specific models. Similar to our architecture, [23] exploits the encoder-decoder transformer blocks to bottleneck clips into a few "grade-aware" output embeddings. Our focus is on interpretability, so we fix the number of output embeddings to be the number of elements. Rather than computing the final score through a Likert-based rubric, we use the rubric developed by competitive figure skating judges for scoring. [9] use a two-stage approach, where one fully-supervised segmentation model segments the video and then, a second model scores each segment individually. In contrast, we propose a single-stage model where "segmentation" is done implicitly, which does not require additional annotations and is more efficient to train (single-stage).

Interpretability. In action quality assessment, [25] use Grad-CAM saliency maps [18] to highlight regions in a diving frame that impact the score prediction (positively or negatively) and [14] use score gradients to identify directions in which *joints* can be moved to *improve* scores. The former could be difficult to interpret by a figure skating athlete, and the latter relies on pose features rather than visual features so it can't exploit strong pretrained video models. An alternative is to make the model interpretable inherently [16]. For example, ad-hoc interpretability is built into [3] where image parts are mapped to their closest visual *part* prototypes of object categories, and then combined to produce a final object category prediction. [10] uses a vision-language model, CLIP [15], to calculate an alignment score between an image and each object attribute for an object class (text prototypes) and then aggregates the score for image classification. Similarly, we use alignment between element representations and natural language rubric text defined by judges to *score* figure skating videos.

3. Method

Overview. Our focus is on utilizing rubric items that are mostly objective, so we focus on technical element scoring (TES) rather than program component scoring (PCS) which captures more artistic, abstract aspects of performance. TES combines difficulty (base value) and quality (grade of execution) over all the elements. To decouple difficulty and quality, and enable interpretable scoring, we use an element list provided in advance (just like a competition) to focus on a single quality prediction task. To be able to score each element independently, we propose two technical innovations. First, to score without explicit element segmentation, our element transformer produces element embeddings using regularization techniques to guide attention toward element segments. Second, implicitly aligning rubric items' text with element embeddings through score regression is difficult with limited data, so we use visual-only rank-aware and vision-text pretraining to align high-quality and low-quality elements with rubric items. Our method aligns model cross-attentions with human expectations and rubric guidelines.

3.1. Decouping difficulty and quality

We focus on the figure skating short program, where skaters perform a series of seven elements within a specified time limit. The *difficulty* is captured by the *base value* (BV), a score predetermined by the ISU for each element type. For example, a "double axel" requiring two rotations in the air during an axel jump would be awarded 3.3, and a "triple axel" with increased difficulty due to an extra rotation would be awarded 8.0. The *quality* (*grade of execution*, GOE) focuses on aspects such as the correctness of form, the takeoff leg, and landing stability, along with other aspects like speed, control, and overall fluidity of the movement. If a severe mistake impacts the perceived difficulty, the BV is deducted by some points (e.g., drifting on the ice during a spin), or the element is downgraded to an easier element (e.g., triple axel to double axel). The routine is shared in advance to judges so a base value score is computed and only deductions and GOE need to be scored after each element's performance. Figure skating datasets provide only the technical element (TES) score rather than the entire score sheet that lists the per-element score breakdown of difficulty (BV) and quality (GOE).

We look up the base value from our knowledge base (ISU protocol) for each element and combine this with a neural model to predict the grade of execution per element and this is regressed against the ground truth technical element score of the performance. Base values can be found in two places: the ISU scoring cheatsheet[1], or scoring sheets[2]. Base values from score sheets include deductions/downgrades and this provides an unfair advantage compared to an element list per performance and *static, performance-agnostic* base value lookup table (without deductions/downgrades). Thus we use the ISU scoring cheatsheet to scrape the base values corresponding to each element, agnostic to a particular performance. In one experiment, we also compare base value reported in the scoring sheet like [9] to using base values from the ISU scoring cheatsheet, but we do not use scoring sheets for our method. We also scrape the grade of execution for visual-only rank-aware pretraining, explained later.

[1] https://usfigureskating.org/sites/default/files/media-files/Scoring%20Cheat%20Sheet.pdf

[2] https://skatingscores.com/

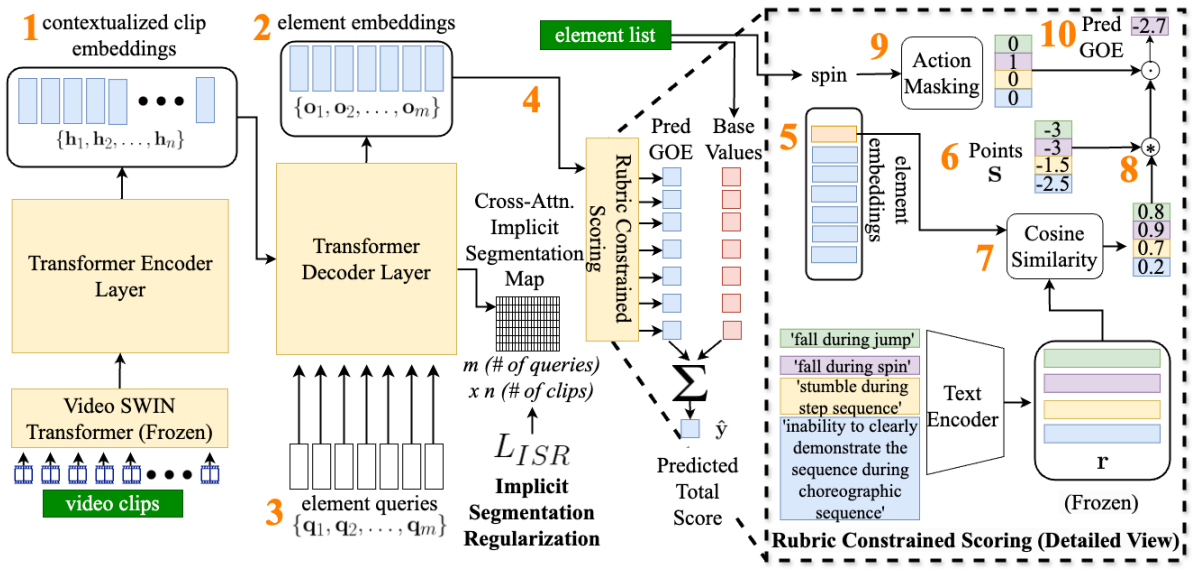

Figure 2. Overview of our proposed architecture. The video and element list (in dark green) of the routine are inputs to our framework. The transformer encoder-decoder architecture produces contextualized clip embeddings (1) from the encoder and element embeddings (2) from the decoder. The latter is produced from just clip embeddings without additional element segmentation annotations through learnable element queries (3) and cross-attention. The element embeddings are input (4) to the rubric-constrained scoring head. In the detailed view of the rubric-constrained scoring head (right), we follow the intermediate outputs corresponding to the highlighted element (5) and only a shortened rubric list is shown, listing some negative criteria (full list in supp.) and their associated points (6). To score, the element embedding is evaluated against negative criteria via cosine similarity (7) to weigh each criterion's contribution (8) towards the predicted GOE. Then, action masking (9) uses the element name to avoid using irrelevant criteria to compute the GOE (10).

Element transformer. Figure skating short programs consist of seven elements with different types of jumps and spins, and one step sequence or choreographic step sequence placed throughout the 2-3 minute program. Since we know the number of elements but don't have ground truth segmentation for where the elements occur in the video, we utilize queries in the transformer decoder layer to implicitly localize and produce element embeddings which are then used to score each element. First, the element transformer encoder takes frozen clip feature embeddings $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ where n is either fixed (when batched) or the maximum number of non-overlapping consecutive clips (inference) in an input video. The encoder then produces contextualized clip embeddings, $\mathbf{H} = \text{Encoder}(\mathbf{E}) = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$. Latent queries $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}$ in the element transformer decoder are used to bottleneck the contextualized clip embeddings $\mathbf{H}$ into m element embeddings, $\mathbf{O} = \text{Decoder}(\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}, \mathbf{H})$, where m are the number of elements (7 in short program). The dimensions of e, h, q, o are $1024, 768, 768, 768$, respectively. Note, the dimension for e depends on the pretrained clip encoder (Video SWIN [8]). This approach is inspired by DETR [2] which uses a transformer encoder-decoder architecture for detection. DETR found that the latent queries in the decoder operated like object anchors, attending to different regions in the image to recognize and detect possible objects. Similarly, in our case, each "anchor" should attend to each element in the video.

Predicting GOE scores. After getting element embeddings, we apply an MLP scoring head, f, to produce a grade-of-execution score; this is combined with the base value of the element for the final score. We denote the base value lookup table as BVL and $BVL(j)$ looks up the base value for the j^{th} element in the program (without deduction information) and i is the i^{th} video. The predicted score, $y_i \in \mathcal{R}$, is then regressed via the mean squared error loss.

$$\hat{y}_i = \sum_{j}^{m} BVL(j) + f(o_i^j) \quad (1)$$

$$L_{se} = (y_i - \hat{y}_i)^2 \quad (2)$$

3.2. Implicit segmentation regularization (ISR)

Implicit segmentation regularization introduces element segmentation priors for the cross-attention maps in the final decoder layer. Cross-attention at each decoder layer is computed between queries, $\mathbf{Q}$, and the contextualized clip embeddings, $\mathbf{H}$, generated by the encoder. This cross-attention distribution over clips softly segments the video into clips most significant for a specific query, q_j, and is used to compute a weighted average of the clip embeddings for each query. Thus, the cross-attention distribution will directly in-

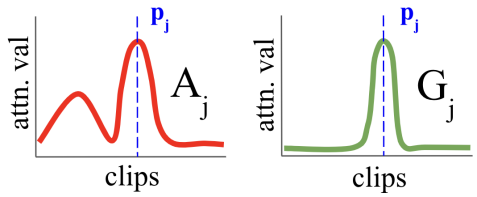

Figure 3. For each query, j, we greedily select the peak p_j with the largest attention value from cross-attention distribution (A_j) to compute the desired attention distribution (G_j). Then, A_j is encouraged to align with G_j in the peak loss.

fluence the element embeddings (output of decoder). However, we noticed that without priors, the queries attend to the same region in the video (redundant) and to temporally disjoint clips of the video (non-contiguous). To address these issues, we propose two regularization techniques: enforcing orthogonality between cross-attention distributions from different queries to reduce redundancy and encouraging peakiness in the cross-attention distribution to ensure attention is focused on specific, contiguous clips.

First, the orthogonality loss reduces redundant attention across queries. Given the cross-attention map between the latent queries and the contextualized clip embeddings $\mathbf{H}$ is $\mathbf{A} \in \mathcal{R}^{m \times n}$, we minimize the dot product between cross-attention distributions $\mathcal{R}^n$ from different latent queries:

$$L_o = \sum_{k=0}^{m} \sum_{j=0}^{m} \mathbf{A}_k \cdot \mathbf{A}_j \quad \text{for} \quad k \neq j \tag{3}$$

L_o will encourage each query to attend to a different region in the video as the minimum dot product (0) indicates no overlap between cross-attention distributions.

Second, we encourage the cross-attention distributions to be concentrated in contiguous clips in videos to satisfy the contiguous aspect of an element segment. This is done by generating a desired distribution that maximizes attention over a single set of contiguous clips. Without annotations on element segments, we must use a heuristic to generate this desired distribution. In Fig. 3, we demonstrate how from the cross-attention distribution A_j, we select the clip (peak) $p_j \in [0, n]$ with the maximum attention value. This peak is the center of a desired attention distribution, modeled as a normal distribution. To align the actual attention and desired attention distributions, we minimize the KL divergence between the two.

More formally, let $\mathbf{A}_j \in \mathbb{R}^n$ be the cross-attention distribution for the j-th query. We aim for this attention distribution to be normally distributed around the selected peak, p_j. So, the desired attention $\mathbf{G}_j^i \in \mathcal{R}$ probability over the ith clip for query j is defined as:

$$G_j^i = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(i - p_j)^2}{2\sigma^2}\right) \tag{4}$$

where σ controls the spread of the distribution (set to 1.5, but could be conditioned on the type of element).

The peak loss is:

$$L_p = \sum_{j=1}^{m} D_{\text{KL}}(\mathbf{A}_j \| \mathbf{G}_j) = \sum_{j=1}^{m} \sum_{i=1}^{n} A_j^i \log\left(\frac{A_j^i}{G_j^i}\right) \tag{5}$$

This will encourage $\mathbf{A}_j$ to attend to a few, contiguous clips rather than disjoint segments over the video.

The final loss is a combination of the regularization losses and the regression loss (squared error):

$$\mathcal{L}_{ISR} = L_o + L_p \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{se} + \mathcal{L}_{ISR} \tag{7}$$

3.3. Rubric-constrained scoring

Having decoupled difficulty from quality and produced element-level representation through the element transformer and regularization, we can now apply the *quality-only* rubric items for scoring *each element*.

Rubric items. For figure skating assessment, judges have a detailed rubric that indicates when to award points and when to deduct points. This rubric contains positive rubric text like "spin with good speed or acceleration during spin" or negative rubric text like "fall during jump". More rubric text examples are shown in Fig. 2 and the full list is in supp. Generally, the rubric items can be divided into two dimensions: sentiment (positive points vs. negative points) and the super-action category (jump, spin, step sequence). While positive rubric items have an equal weight (+0.5), the negative rubric items have deductions at differing levels of severity as shown by points (numbered 6) in Fig. 2.

Scoring. For rubric-constrained scoring, the predicted grade of execution is computed by weighing the points for each rubric item according to how much an element aligns with that rubric item. Specifically, a pretrained text encoder computes text embeddings for each rubric item; in our experiments, we use the CLIP [15] text encoder. Then, alignment is computed using the cosine similarity between the element embeddings $\mathbf{o}_i$ and rubric text embeddings $\mathbf{r}$.

$$\cos(\mathbf{o}_i^j, \mathbf{r}_k) = \frac{\mathbf{o}_i^j \cdot \mathbf{r}_k}{\|\mathbf{o}_i^j\| \|\mathbf{r}_k\|} \tag{8}$$

The cosine similarity (rubric activations) is then used to generate a weighted score; the weighted score for the j-th element embedding and the k-th rubric item with an associated score of s_k is:

$$w_{ij} = s_k \cdot \cos(\mathbf{o}_i^j, \mathbf{r}_k) \tag{9}$$

This scoring mechanism replaces the MLP regressor in Eq. 1. This reduces the number of trainable parameters and enhances interpretability, as the rubric activations directly

contribute to the grade of execution and total score. Additionally, the rubric activations are linked to the rubric's natural language text, making this scoring process *understandable* to a human compared to a black-box MLP. The final grade of execution is the sum of the weighted scores across all rubric items for each element and then sigmoid rescaled within the grade of execution range, $[-3, 3]$:

$$\text{GOE}_i^j = 6 * \sigma(\sum_k s_k \cdot \cos(\mathbf{o}_i^j, \mathbf{r}_k)) - 3 \quad (10)$$

This requires no additional changes during inference to predict the final score, just like Eq. 1.

Action masking. Not all rubric items are relevant to scoring; for example, if "triple axel" is performed then only the jump rubric items (positive or negative) are relevant. For this, we propose masking rubric activations from rubric items not corresponding to the super-action, so that they don't contribute to the predicted grade of execution.

Pretraining for video-rubric text alignment. Initially, there is low alignment between element embeddings with rubric text embeddings, leading to suboptimal solutions with rubric activations close to zero. To address this, we use a contrastive triplet loss [17] to align the embeddings. Since rubric breakdowns are unavailable, we heuristically assume that high-scoring elements align more with positive rubric items and low-scoring elements align more with negative rubric items. Using the grade of execution scores, we form triplets: a high-scoring element embedding as the anchor, o_a, with sampled positive r_a^+ and negative r_a^- rubric text embeddings, and vice versa for low-scoring elements. The triplet loss is then applied to these sets:

$$\mathcal{L}_{\text{vt}} = \sum_a \left[\max\left(0, d(\mathbf{o}_a, \mathbf{r}_a^+) - d(\mathbf{o}_a, \mathbf{r}_a^-) + \mathbf{m}\right)\right] \quad (11)$$

where a is from the set of indices corresponding to high-scoring elements (above a threshold, 1.3) and low-scoring elements (below a threshold, -2), and $\mathbf{m}$ is the margin. We denote this visual-text pretraining PT_{vt} in experiments.

We selected triplet loss over other contrastive losses which require large batch sizes, as it only needs one positive and one negative which keeps the computational demands low. We tried other losses like cosine embedding loss, but found triplet loss more stable to train.

Additional visual pretraining. Since figure skating scoring datasets are fairly small, we use contrastive pretraining within the visual space to enforce geometric structure that is rank-sensitive [4]. For example, high-scoring samples of the same super-action should be closer in the visual embedding space than low-scoring samples of the same super-action, and vice versa. To do this, we also use a triplet loss. In this case, we also treat the element embedding as an anchor $\mathbf{o}_a$ but without any grade of execution (GOE) threshold restrictions compared to $\mathcal{L}_{\text{vt}}$ above. Per batch, for each element a, in the batch, a positive pair, $\mathbf{o}_a^+$, is selected on the basis of the super-category and difficulty being exact matches and the GOE within 0.2 between the pair. A close negative, $\mathbf{o}_a^-$, is selected for the element a where both the super-category and difficulty match, but the GOE differs by more than 0.5.

$$\mathcal{L}_{\text{vis}} = \sum_a \left[\max\left(0, d(\mathbf{o}_a, \mathbf{o}_a^+) + \mathbf{m}\right)\right] \quad (12)$$

We denote this visual pretraining PT_{vis} in experiments.

We combine these two pretraining objectives and train them jointly by alternating between $\mathcal{L}_{vt}$ and $\mathcal{L}_{vis}$ each epoch. This is denoted as PT_{joint} in our experiments.

4. Experiments

We will first show the value of doing a base value lookup as explained in Sec. 3.1 and our full method and compare them against prior state-of-the-art methods. Then, we ablate each aspect of our method. Lastly, we evaluate the impact of each component of our method on implicit segmentation. We do not apply our method to Rhythmic Gymnastics (RG) [26] since certain aspects of our method, such as scoring each element independently, implicit element segmentation, action masking, and vision-text pretraining (rubric contains only negatives), are not directly applicable. We leave these challenges for future work.

Experiment details. We use the Fis-V dataset [24] for our experiments. This consists of 400 training videos and 100 testing videos. On average, videos are 2-3 minutes long. These are from broadcasts of international skating competitions scraped between the years 2014-2017. The grade of execution scoring is between [-3,3] instead of the post-2018 modern range of [-5,5]. We follow [23] and sample non-overlapping 16-frame clips spanning the entire video and extract frozen embeddings for each clip from Video Swin Transformer [8]. There is one encoder layer and two decoder layers. For our implementation, we use PyTorch 1.10.1 with CUDA 10.2 and train on a single Quadro RTX5000 GPU. During training, we use a batch size of 64 and a learning rate of 1e-5. We use a batch size of 32 and a learning rate of 1e-2 during pretraining. For both, we use a fixed number of clips, 128, and during inference, any number of clips can be provided as input. Lastly, we use RMSProp optimizer and weight decay of 1e-4.

Metrics. (1) For our experiments measuring the scoring ability of models, we report mean-squared error (MSE), which judges the precision of the predicted score, and Spearman correlation, which judges the relative rank of multiple scores. There are two use cases for an athlete, one is when practice needs to be evaluated without reference (no other performers), then score precision, MSE, is more important in selecting the best model. However, if the athlete wanted to compare multiple performances of their own, the

Method	MSE (↓)	Sp. Corr. (↑)
CoRe** [25] (2021)	23.50	0.66
GDLT* [23] (2022)	33.60	0.69
TPT** [1] (2022)	27.50	0.57
MLP-Mixer** [21] (2023)	19.57	0.68
SGN [6] (2024)	<u>19.05</u>	0.70
Base Value Lookup (BVL)	19.53	0.76
GDLT (2022) [23] w/ BVL	28.52	<u>0.77</u>
Ours	**9.34**	**0.84**
GT Base Value	12.03	0.91

Table 1. Comparison of our proposed method (Ours), a strong baseline (Base Value Lookup), and state-of-the-art methods on FIS-V dataset [24] technical execution score (TES) component. Note [9] is not reported as it was applied only to MIT-Skate [14], a smaller dataset and was trained on private segmentation annotations, thus not reproducible. Bold indicates best in column; underline indicates second-best in column. * indicates our reproduced results. ** indicates reproduced by [6].

precision can be lower as long as there is relative consistency where a better performance receives a higher score than a worse performance regardless of the specific score. (2) We also evaluate how much the cross-attention peak per query corresponds to elements. We annotate 56 elements in videos sampled from Fis-V [24] with start and end timestamps. We use the metric of order-sensitive precision calculated by the ratio between the number of cross-attention peaks that fall into the correct segment corresponding to the query/element. This order-sensitive precision can be too strict so we have two other metrics: order-insensitive precision and order-insensitive precision (1:1). The latter enforces a 1:1 assignment between peaks and the matched ground truth segment whereas the former is less strict and multiple peaks can be matched to the same segment. 1:1 assignment is determined by using the assignment permutation that yields the highest matches between the peak and ground truth segment. The less-strict metric considers if the query attends to *at least* an element in the sequence rather than the background which is the minimum level of intuitiveness for scoring.

4.1. State-of-the-art comparison

Just like how judges receive the sequence of elements for the routine, our method requires an element list per video. We use this to lookup the base value (represents *difficulty*) per element name. The Base Value Lookup *baseline* takes the sum over the mapped base value for each element in the sequence as the "predicted score". In Tab. 1, we observe that prior methods fall significantly behind on Spearman correlation compared to this simple base-value lookup that doesn't even consider quality.

In prior work [9], base values are scraped from ground-truth score sheets, however, these include deductions (indicators of *quality*) or downgrades if the element performed is missing revolutions. In Tab. 1 we see that the use of ground truth base value like in [9] provides an unfair advantage as shown by the superior performance of using the ground truth base value in the last row compared to all the other prior state of the art methods in MSE and all methods, including ours, when comparing the Spearman correlation; this shows that the ground truth base value from the score sheets does not separate difficulty and quality, rather quality also entangled with the reported base value. So, our Base Value Lookup is more fair to provide and is also truly known in advance. Our full method improves significantly over the Base Value Lookup.

Method	MSE (↓)	Sp. Corr. (↑)
ElTr w/ Orig. Rubric	11.87	0.816
ElTr w/ S. Rubric	**10.68**	**0.828**

Table 2. Effect of simplifying rubric. ElTr = Element Transformer, S.Rubric = Simplified Rubric. Bold indicates best in column.

Type	HN	MSE (↓)	Sp. Corr. (↑)
Org.	No	13.99	0.810
Simpl.	No	**12.27**	<u>0.811</u>
Simpl.	Yes	<u>13.03</u>	**0.815**

Table 3. Rubric type and use of hand negatives during visual-text pretraining; reporting performance on Fis-V test after fine-tuning (without action masking) on train split.

4.2. Ablations

We show the influence of augmenting the rubric text on scoring performance, and show that better text inputs lead to better performance. We ablate implicit segmentation regularization and show the effectiveness on both scoring precision and implicit segmentation. We also compare other important aspects of our method like using hand negatives during visual-text pretraining and action masking.

Simplified rubric text. Some rubric items are verbose and use terminology not easily understood ("element combo of one jump, final goe must be"); this also impacts the quality or sentiment (positive or negative) separation and element type separation of rubric items in the text embedding space. For more well-separated text representation, we simplified the rubric to avoid using specialized terms and long sentences (denoted as S. Rubric). In Tab. 2, we see that using simplified rubric text on top of the element transformer, without pretraining, improves both MSE and Sp. Corr. The rubric text may be more informative and clearer, leading to better text embedding representation.

Pretraining. We ablate the margin in both visual pretraining and visual-text pretraining (tables in supp), and we

Method	MSE (↓)	Sp. Corr. (↑)
Element Transformer	13.14	0.824
+ $\mathcal{L}_{peak}$	<u>11.50</u>	0.817
+ $\mathcal{L}_{ortho}$	12.02	<u>0.825</u>
+ $\mathcal{L}_{ISR}$	**10.68**	**0.828**

Table 4. Regularization ablations without pretraining. ISR = implicit segmentation regularization.

Method	MSE (↓)	Sp. Corr. (↑)
Ours	**10.68**	**0.828**
Our w/o AM	14.68	0.785

Table 5. Effect of action masking. These experiments are without pretraining. AM = Action Masking. Bold indicates best in column.

find the optimal margin parameters with 0.5 for PT_{vis} and 1.0 for PT_{vt} and use these settings for PT_{joint} and achieve our best MSE (9.34) and correlation (0.84). In Tab. 3, we also experiment with crafting hand negatives for each rubric item (the hand negative for "combo contained only one jump" is "combo contained two jumps"; the full list can be found in supp.). Simplifying the rubric had a positive impact during visual-text pretraining, but hand negatives had mixed results. We still use hand negatives in our final joint pretraining method. We show more results in supp.

Effect of implicit segmentation regularization on score prediction. In Tab. 4, we observe that the orthogonality loss lowers mean-squared error, but degrades Spearman correlation compared to the vanilla element transformer; peak loss improves on both metrics. Combining these two losses as implicit segmentation regularization (ISR) yields lower mean-squared error and improves Spearman correlation. Beyond metrics, when looking at a sample cross-attention map (shown in supp), we observe that the element transformer with no regularization has severe redundancy in the queries; all queries have the same cross-attention pattern. However, after ISR, the queries have more obvious peaks and focus on disjoint parts of the video, as desired.

Action masking. We also ablate action masking during rubric-constrained scoring in Tab. 5. We find that action masking is especially important for improving score precision, and also improves Spearman correlation. We suspect the scoring precision improves by eliminating GOE contribution from irrelevant criteria.

Accuracy of implicit segmentation and impact of pretraining. We observe in Tab. 6 that peak loss improves order-sensitive precision but overall the queries do not attend to elements, instead background. The orthogonality loss improves the ability to attend to actual elements as evidenced by serious jumps in order-insensitive metrics (35.7 to 57.1 and 12.5 to 32.1); however the order-senstive precision drops to zero, indicating that the element query is

Method	$\mathbf{P}_{OIS}$ (%)	$\mathbf{P}_{OIS}$ (1:1) (%)	$\mathbf{P}_{OS}$ (%)
Ours w/o $\mathcal{L}_{ISR}$	35.7	12.5	0.0
+ $\mathcal{L}_{peak}$	25.0	3.6	3.6
+ $\mathcal{L}_{ortho}$	57.1	32.1	0.0
+ $\mathcal{L}_{ISR}$	39.3	33.9	7.1
+ $\mathcal{L}_{ISR}$, PT_{vis}	42.9	37.5	3.6
+ $\mathcal{L}_{ISR}$, PT_{vt}	67.9	42.9	1.8
+ $\mathcal{L}_{ISR}$, PT_{joint}	58.9	35.7	1.8

Table 6. Effect of pretraining and regularization on implicit segmentation ability. The "attention peak" over clips for each element query is the clip with the maximum cross attention value for that query. P_{OIS} = Order-Insensitive (OIS) Precision which is correct when the attn. peak overlaps with any annotated element segment in the video. P_{OIS} (1:1) = a modified version of P_{OIS} where a 1:1 assignment between attn. peaks and annotated element segments of a video is enforced. P_{OS} = Order-Sensitive (OS) Precision which is correct when the attn peak overlaps with corresponding annotated segment based on element query position.

not attending to element corresponding to its order. When combining these regularization methods, there is an improvement in order-sensitive precision. In terms of pretraining, visual-text pretraining seems to have the most impact on order-insensitive metrics compared to the other method variants. Joint pretraining improves both order-insensitive precision metrics compared to our vanilla method. However, there is a drop in order-sensitive precision. These results indicate that while there is an improvement in the explainability (at least each query attends to an actual element), due to the lack of explicit segment supervision, the queries don't attend to elements in the correct order. Further research is needed to address this.

5. Conclusion

We showed a new, interpretable, well-performant mechanism for action quality assessment in figure skating. It relies on a well-defined rubric of criteria for figure skating elements, and an implicit segmentation approach to obtain element-level scoring. Our approach is a first step in using freely available, structured, language-based resources for improving interpretability in figure-skating scoring. In the future, we will experiment with further peak loss and ordering loss techniques. Adding explicit element supervision will likely improve performance and explainability, which opens the possibility of actual use from athletes.

Acknowledgement. This work was supported by National Science Foundation Grants No. 2006885 and partially by the University of Pittsburgh Center for Research Computing, RRID:SCR_022735, through the resources provided. Specifically, this work used the H2P cluster, which is supported by NSF award number OAC-2117681.

References

[1] Yang Bai, Desen Zhou, Songyang Zhang, Jian Wang, Errui Ding, Yu Guan, Yang Long, and Jingdong Wang. Action quality assessment with temporal parsing transformer. *European Conference on Computer Vision*, 2022. 7

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*, volume 12346 of *Lecture Notes in Computer Science*, page 213–229. Springer International Publishing, Cham, 2020. 4

[3] Chaofan Chen, Oscar Li, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *Neural Information Processing Systems*, 2018. 3

[4] Hazel Doughty, Dima Damen, and W. Mayol-Cuevas. Who's better? who's best? pairwise deep ranking for skill determination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6057–6066, 2017. 6

[5] Hazel Doughty, W. Mayol-Cuevas, and Dima Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7854–7863, 2018. 2

[6] Zexing Du, Di He, Xue Wang, and Qing Wang. Learning semantics-guided representations for scoring figure skating. *IEEE Transactions on Multimedia*, 26:4987–4997, 2024. 3, 7

[7] Yongjun Li, Xiujuan Chai, and Xilin Chen. End-to-end learning for action quality assessment. In *Pacific Rim Conference on Multimedia*, 2018. 2

[8] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3192–3201, 2021. 4, 6

[9] Hitoshi Matsuyama, Nobuo Kawaguchi, and Brian Y. Lim. Iris: Interpretable rubric-informed segmentation for action quality assessment. *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2023. 3, 7

[10] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *ICLR*, 2023. 3

[11] Paritosh Parmar and Brendan Tran Morris. Learning to score olympic events. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 76–84, 2016. 2

[12] Paritosh Parmar and Brendan Tran Morris. Action quality assessment across multiple actions. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1468–1476, 2018. 2

[13] Paritosh Parmar and Brendan Tran Morris. What and how well you performed? a multitask learning approach to action quality assessment. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–313, 2019. 3

[14] Hamed Pirsiavash, Carl Vondrick, and Antonio Torralba. Assessing the quality of actions. In *European Conference on Computer Vision*, 2014. 1, 2, 3, 7

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 3, 5

[16] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206 – 215, 2018. 3

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 6

[18] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336 – 359, 2016. 3

[19] Yansong Tang, Zanlin Ni, Jiahuan Zhou, Danyang Zhang, Jiwen Lu, Ying Wu, and Jie Zhou. Uncertainty-aware score distribution learning for action quality assessment. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9836–9845, 2020. 2

[20] Tianyu Wang, Yijie Wang, and Mian Li. Towards accurate and interpretable surgical skill assessment: A video-based method incorporating recognized surgical gestures and skill levels. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020. 3

[21] Jingfei Xia, Mingchen Zhuge, Tiantian Geng, Shun Fan, Yuantai Wei, Zhenyu He, and Feng Zheng. Skating-mixer: Long-term sport audio-visual modeling with mlps. In *AAAI Conference on Artificial Intelligence*, 2022. 2, 7

[22] Xiang Xiang, Ye Tian, Austin Reiter, Gregory Hager, and Trac D. Tran. S3d: Stacking segmental p3d for action quality assessment. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 928–932, 2018. 2

[23] Angchi Xu, Ling-An Zeng, and Weihao Zheng. Likert scoring with grade decoupling for long-term action assessment. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3222–3231, 2022. 3, 6, 7

[24] C. Xu, Yanwei Fu, Bing Zhang, Z. Chen, Yu-Gang Jiang, and X. Xue. Learning to score figure skating sport videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:4578–4590, 2018. 2, 6, 7

[25] Xumin Yu, Yongming Rao, Wenliang Zhao, Jiwen Lu, and Jie Zhou. Group-aware contrastive regression for action quality assessment. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7899–7908, 2021. 2, 3, 7

[26] Ling-An Zeng, Fa-Ting Hong, Wei-Shi Zheng, Qi-Zhi Yu, Wei Zeng, Yao-Wei Wang, and Jian-Huang Lai. Hybrid dynamic-static context-aware attention network for action assessment in long videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 2526–2534, New York, NY, USA, 2020. Association for Computing Machinery. 6