1 Supplementary Material

1.1 Complete Hyperparameter Specifications

Table 1 provides the complete set of hyperparameters used in our experiments.

1.2 Implementation Details

Hardware and Software. All experiments were conducted on a single NVIDIA A100 GPU (40GB VRAM) with AMD EPYC 7763 CPUs. We used PyTorch 2.1.0, CUDA 12.1, and Python 3.10. Mixed-precision training (FP16) was employed to reduce memory usage. Each model was trained across 5 random seeds, and we report mean performance with standard deviation.

Training Protocol. Models were trained end-to-end on a single GPU with batch size 8. The V-JEPA backbone was frozen after initialization with pretrained weights from [1]. The FUTR decoder and grammar integration layers were trained from scratch. Training took approximately 12 hours per dataset. We used early stopping based on validation accuracy with patience of 10 epochs.

Grammar Induction Details. Grammars were induced offline using the entire training set of each dataset. For CholecT50, we extracted 16,295 annotated sequences. The n-gram mining was performed using a greedy left-to-right longest-match algorithm. Chirality pairs were identified using semantic similarity between verb embeddings (Word2Vec trained on surgical reports) and temporal co-occurrence patterns.

Baseline Implementations. All baselines (RNN, Temporal Aggregation, Cycle Consistency, FUTR, KARI) were implemented using official codebases where available, or faithful re-implementations following the original papers. Hyperparameters for baselines were tuned using grid search on the validation set. We report the best-performing configuration for each baseline.

1.3 Additional Experimental Details

Dataset Preprocessing. Videos were resized to 224×224 resolution and normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). Frame sampling rate was 1 FPS for all datasets. For CholecT50 and SAR-RARP50, triplet annotations (instrument, verb, target) were used. For JIGSAWS and Cholec80, gesture/phase labels were mapped to our unified action vocabulary.

Evaluation Metrics. **Mean-over-Classes (MoC) Accuracy:** We compute per-class accuracy and average across all classes to handle class imbalance. **Chirality Error Rate (CER):** Percentage of predictions where both actions in a chiral pair are predicted simultaneously. **Grammar Validity Score (GVS):** Percentage of predicted sequences that conform to the induced grammar.

Statistical Testing Protocol. For each comparison, we computed per-sample accuracies across the test set. Paired t-tests were conducted using scipy.stats.ttest_rel with alternative='greater' (one-tailed). Wilcoxon signed-rank tests used scipy.stats.wilcoxon. Effect sizes (Cohen's d) were computed as $d = \frac{\mu_{\text{diff}}}{\sigma_{\text{diff}}}$ where μ_{diff} and σ_{diff} are the mean and standard deviation of per-sample accuracy differences. Bonferroni correction was applied for per-verb analyses: $\alpha_{\text{corrected}} = 0.05/10 = 0.005$.

1.4 Runtime and Computational Cost

Inference Speed. Mean inference time per sample: 120ms (P95: 145ms, P99: 162ms). This translates to 8.3 FPS, well above the 1-5 FPS requirement for real-time surgical video processing. Grammar parsing adds negligible overhead (<5ms) due to the compact grammar structure.

Memory Requirements. Peak GPU memory during training: 18.2 GB (with batch size 8). During inference: 1.2 GB. CPU memory for grammar storage: 2 MB. Total model parameters: 87M (V-JEPA: 85M frozen, FUTR decoder: 2M trainable).

Grammar Induction Cost. Offline grammar induction from 16,295 sequences (CholecT50): 3.2 minutes on a single CPU core. The induced grammar file is 1.8 MB (JSON format). Grammar loading time at inference: <100ms.

1.5 Code and Data Availability

Code, trained models, and the induced grammars will be made publicly available here. The CiSA benchmark (chirality-augmented CholecT50 and SAR-RARP50) will also be released with detailed annotation protocols.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985* (2025).

Table 1: Complete hyperparameter configuration for PTG training and inference.

Component	Parameter	Value
Training	Optimizer	AdamW
	Learning rate	1×10^{-3}
	Weight decay	1×10^{-4}
	Batch size	8
	Epochs	60
	Warmup epochs	10
	LR schedule	Cosine annealing
	Gradient clipping	1.0
V-JEPA Backbone	Model	vjepa2-vitl-fpc64-256
	Embedding dim	1024
	Frames per clip	{8, 16, 64}
	Frozen	Yes
	Pooling	Temporal average
FUTR Decoder	Hidden dimension	512
	Number of layers	6
	Attention heads	8
	Feedforward dim	2048
	Dropout	0.1
	Activation	GELU
Loss Weights	Action loss (L_{action})	1.0
	Duration loss (λ_{dur})	0.5
	Object loss (λ_{obj})	0.3
	Goal loss (λ_{goal})	0.2
	Grammar KL (λ_{gram})	0.3
	Object consistency (λ_o)	0.4
	Goal reward (λ_r)	0.3
Grammar Priors	Temperature (τ)	0.5
	Chirality boost (λ_{chiral})	0.15
	Min n-gram frequency	5
	Max n-gram length	4
	Markov order	2
	Duration model	Laplace (median+MAD)
Data Augmentation	Horizontal flip	0.5 prob
	Random crop	0.8 prob
	Color jitter	0.3 prob
	Temporal jitter	±2 frames
Observation Ratios	Training α	{0.2, 0.3, 0.5, 0.7, 0.9, 1.0}
	Inference α	{0.2, 0.3}
	Prediction horizons β	{0.1, 0.2, 0.3, 0.5}
	Future timesteps T	5