# SEMAR: SEMANTIC MARGINAL-VALUE ROUTING FOR BUDGET-ADAPTIVE SIGN LANGUAGE VIDEO REFINEMENT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

An important sign-language region may already be correct, while a visible error may not improve with further refinement. We study compute allocation through the *Semantic Value of Compute* (SVoC): the expected semantic-utility gain of an executable refinement increment per unit of measured hardware cost. SEMAR obtains paired marginal-gain labels from shared-prefix trajectories, predicts gains for articulator–time tubes and refinement depths, and selects increments under a packed-plan latency budget. In matched-backbone evaluation at 50% residual budget, SEMAR achieves 20.1 WER at 682 ms P95, compared with 27.8 for uniform routing, 21.4 for predicted error reduction, and 21.0 for predicted cumulative gain (direct) at comparable latency. Relative to unconditional full refinement, it reduces P95 latency by 37% with a 2.8-point WER gap. Shared-prefix supervision reduces marginal-label variance by 53%. Experiments across DGS and ASL, signer-disjoint data, refiner families, and three GPUs support semantic marginal-value routing as a practical quality–latency trade-off.

## 1 INTRODUCTION

Sign languages express meaning through coordinated manual and non-manual articulators. A visually plausible video can still contain an incorrect handshape, orientation, contact, mouth gesture, or eyebrow movement. Neural sign-language production (SLP) has progressed from text/gloss-to-pose pipelines and rendering Stoll et al. (2018); Saunders et al. (2020) to motion primitives, co-articulation, and diffusion-based synthesis Saunders et al. (2021; 2022); Baltatzis et al. (2024); Xie et al. (2024); Ma et al. (2024). These advances improve generation, but leave a separate decision: where should a limited refinement budget be spent?

A fixed hand/face rule captures the importance of local articulators but not the return of refining them. Two equally important hand tubes may differ in current correctness and recoverability: one is already correct, while the other contains an error that another refinement increment can repair. Conversely, a large error may persist under the available refiner. Allocating by importance or current error alone therefore need not select the most useful computation.

We formulate the routing target as the *semantic improvement caused by an executable refinement increment*. For each tube and depth, SVoC divides expected semantic-utility gain by the increment's measured cost. This action-level target connects the current generated state to the benefit of extending its refinement trajectory, as illustrated in Fig. 1.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

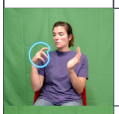
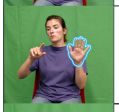
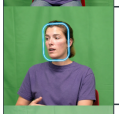
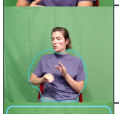
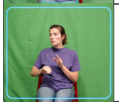
| Candidate | Current Status | Marginal Gain | Hardware Cost | SVoC | Decision |
|---|---|---|---|---|---|
|  | • Handshape error<br>• Recoverable | HIGH | LOW | HIGH | Refine |
|  | • Correct | LOW | LOW | LOW | Skip |
|  | • Limited expected improvement | LOW | HIGH | Low | Skip |
|  | • Motion Error<br>• Unrecoverable | LOW | MEDIUM | Low | Skip |
|  | • Semantically irrelevant detail | LOW | MEDIUM | LOW | SKIP |

Figure 1: Semantic importance alone does not determine where additional computation should be allocated. For each candidate tube and refinement increment, SEMAR predicts marginal semantic gain and normalizes it by measured incremental hardware cost. The allocator prioritizes recoverable errors with high semantic gain per unit cost while skipping important-but-correct and low-return regions.

SEMAR adds a budget-adaptive routing layer to a fixed coarse-to-fine generator. A coarse autoregressive model produces the video latent; a tube head proposes articulator–time regions. During label construction, a frozen semantic evaluator scores adjacent checkpoints of the same residual trajectory. Shared stochastic prefixes pair the two executions. A lightweight Router then learns the raw marginal gains from coarse-state features. At inference, a priority queue ranks available increments by predicted gain per measured cost and admits them using the predicted P95 latency of the complete packed plan.

Our contributions are:

- An action- and depth-specific supervision scheme for semantic marginal gain, constructed from paired checkpoints of executable refinement trajectories.
- A learned marginal-value routing layer with precedence-aware, hardware-budgeted allocation, evaluated through matched-backbone routing controls, independent machine evaluators, and end-to-end system measurements.

## 2 Related Work

**Sign-language generation and refinement.** Neural SLP spans text/gloss-to-pose pipelines, progressive autoregression, motion primitives, latent motion, and diffusion-based synthesis Stoll et al. (2018); Saunders et al. (2020; 2021; 2022); Xie et al. (2024); Fang et al. (2023); Baltatzis et al. (2024); Ma et al. (2024). Recent text-to-video, retrieval/transition, regional, iterative, and hybrid diffusion methods improve global motion and local detail Wang et al. (2025); Zuo et al. (2025); Tang et al. (2025); Shi et al. (2025); Kiziltepe et al. (2025); Ye et al. (2026); Yu et al. (2026). SEMAR addresses the allocation of an existing refiner: its supervision measures the depth-specific semantic return of each candidate action.

**Evaluation.** PHOENIX14T and How2Sign provide DGS and ASL benchmarks Forster et al. (2014); Duarte et al. (2021). Back-translation uses recognition or translation models Camgoz et al. (2018; 2020) and metrics such as BLEU and ROUGE Papineni et al. (2002); Lin (2004); pose-based comparisons often use temporal alignment such as DTW Sakoe & Chiba (1978). SiLVERScore and pose meta-evaluation study semantic representations and evaluator validity Imai et al. (2025);

Jiang et al. (2025). Our training utility combines sentence-level and local articulatory signals, while separate evaluators assess generated outputs.

**Adaptive generative computation.** Diffusion, latent video diffusion, and flow matching provide iterative generative trajectories Ho et al. (2020); Song et al. (2021); Rombach et al. (2022); Blattmann et al. (2023); Lipman et al. (2023); Pooladian et al. (2023). Adaptive computation time, block skipping, token reduction, sparse experts, and mixture-of-depths conditionally allocate computation Graves (2016); Wu et al. (2018); Wang et al. (2018); Ryoo et al. (2021); Rao et al. (2021); Yin et al. (2022); Shazeer et al. (2017); Fedus et al. (2022); Raposo et al. (2024). Our specific contribution is a routing label obtained by executing paired generative refinement actions and measuring their marginal semantic consequence, followed by allocation using target-device cost measurements.

# 3 METHODOLOGY

## 3.1 EXECUTABLE REFINEMENT ACTIONS

Let $X = (x_1, \ldots, x_L)$ be a spoken-language sentence, $s$ an optional signer condition, and $V \in \mathbb{R}^{T\times H\times W\times 3}$ the target video. A fixed autoencoder maps $V$ to $z = E_{\mathrm{vid}}(V)$. A fixed coarse generator produces

$$z^c = G_c(X, s; \theta_c), \qquad \hat{V}^c = D_{\mathrm{vid}}(z^c). \tag{1}$$

The default residual model is a latent-video Diffusion Transformer with 12 Transformer blocks, applied to articulator–time tubes. For each tube, it receives the noisy local residual, the corresponding coarse-latent region, a low-resolution context region, a soft write-back mask, text or gloss features, an optional signer embedding, the articulator identity, tube coordinates, and the solver timestep. Textual features enter through cross-attention; timestep and signer embeddings modulate the Transformer blocks through adaptive layer normalization. The model is trained before marginal-label construction and then frozen throughout label generation, Router training, hardware calibration, and final evaluation.

The refiner exposes $K = 3$ nonzero checkpoints along a fixed eight-network-evaluation trajectory, with $(n_0, n_1, n_2, n_3) = (0, 2, 4, 8)$. For a mask $M$ and checkpoint $k$,

$$z^{(k)} = z^c + M \odot R_\theta^{(n_k)}(z^c, X, s, \epsilon), \qquad \hat{V}^{(k)} = D_{\mathrm{vid}}(z^{(k)}), \tag{2}$$

where $R_\theta^{(0)} = 0$ and $\epsilon$ is the initial stochastic state. Equation (2) uses global-latent notation for a gather–refine–scatter interface: the tube coordinates select the local latent and context regions, the residual model updates that local state, and the resulting residual is returned to the same latent coordinates through the soft mask $M$. The exact crop layout is an implementation choice and is not part of the routing objective. All checkpoints are outputs of one trajectory: the 2-step execution is a prefix of the 4-step execution, which is a prefix of the 8-step execution. Thus the three additional compute increments contain 2, 2, and 4 function evaluations, respectively. Checkpoint decoding provides the videos used for paired supervision; the inference plan executes each selected tube only to its final chosen depth.

For selected increments $\mathcal{A}$, let $\mathcal{J}_\mathcal{A} = \{j : \exists k, (j,k) \in \mathcal{A}\}$ and $k_j = \max\{k : (j,k) \in \mathcal{A}\}$. Tube $j$ has a soft write-back mask $A_j$, defined below. Overlap is normalized at latent location $p$:

$$\widetilde{A}_j(p) = \frac{A_j(p)}{\max\big(1, \sum_{\ell\in\mathcal{J}_\mathcal{A}} A_\ell(p)\big)}. \tag{3}$$

The packed plan produces

$$z_\mathcal{A} = z^c + \sum_{j\in\mathcal{J}_\mathcal{A}} \widetilde{A}_j \odot R_\theta^{(n_{k_j})}(z^c, X, s, \epsilon_j), \qquad \hat{V}_\mathcal{A} = D_{\mathrm{vid}}(z_\mathcal{A}), \tag{4}$$

with one stochastic state $\epsilon_j$ shared across the checkpoints of each tube. Packing determines the execution schedule; Eq. (4) determines residual composition. Our objective is

$$\max_\mathcal{A}\ \mathbb{E}[\mathcal{U}_{\mathrm{sem}}(\hat{V}_\mathcal{A}, X)] \quad \text{s.t.} \quad \widehat{\mathcal{C}}^{\mathrm{P95}}_{\mathrm{pack}}(\mathcal{A}; \mathcal{H}) \le B, \tag{5}$$

where $\mathcal{H}$ is the target hardware and $\widehat{C}^{\mathrm{P95}}_{\mathrm{pack}}$ predicts complete-plan P95 latency. The expectation concerns stochastic refinement; the text and any training reference are fixed for each example.

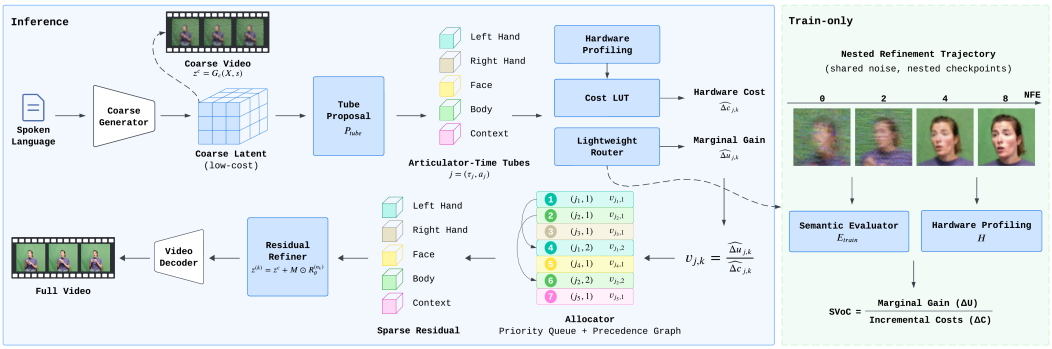



Figure 2: Overview of SEMAR. Counterfactual execution and semantic evaluation are training-only; inference runs the coarse generator, tube proposal, lightweight router, allocator, selected refinements, and decoder.

**Articulator–time tubes.** A decision atom is

$$j = (\tau_j, a_j), \qquad a_j \in \{\mathrm{lh}, \mathrm{rh}, \mathrm{face}, \mathrm{body}, \mathrm{context}\}, \tag{6}$$

with interval $\tau_j = [t_j^s, t_j^e]$ and mask $A_j$. Windows span 16 input frames with a four-frame overlap and are shared across routing policies. OpenPose Cao et al. (2017) supplies pose and face tracks, while MediaPipe Hands Zhang et al. (2020) supplies hand tracks used to provide teacher masks during training. At inference,

$$(\hat{A}_j, o_j, c_j) = P_{\mathrm{tube}}(z^c, X), \tag{7}$$

where $o_j$ represents occlusion/left–right identity and $c_j$ is confidence. Low confidence expands the tube and adds a low-resolution context halo. Predicted masks are used in the same composition rule as teacher masks.

### 3.2 SEMANTIC UTILITY AND PAIRED MARGINAL GAINS

A frozen sign-perception evaluator $E_{\mathrm{train}}$ supplies sentence, gloss, and local articulatory features. Given text $X$ and an available reference $V$, the training utility is

$$\begin{aligned}\mathcal{U}_{\mathrm{sem}}(\hat{V}, X, V) = w_s\ \mathrm{sim}(e_{\mathrm{sent}}(\hat{V}), e_X) - w_g\ \mathrm{CTC}(p_{\mathrm{gloss}}(\hat{V}), g) \\ - w_h \mathcal{D}_{\mathrm{hand}}(\hat{V}, V) - w_o \mathcal{D}_{\mathrm{orient}}(\hat{V}, V) \\ - w_c \mathcal{D}_{\mathrm{contact}}(\hat{V}, V) - w_f \mathcal{D}_{\mathrm{nmm}}(\hat{V}, V).\end{aligned} \tag{8}$$

The local terms measure handshape, orientation, contact, and non-manual discrepancies. Component scales are normalized on validation data and frozen with the utility weights before marginal-label construction. Unavailable terms are omitted and active weights are renormalized per example. PHOENIX14T uses its gloss annotations; How2Sign disables the gloss term unless a verified sequence is available. Reference videos, annotation targets, and $E_{\mathrm{train}}$ are used to construct labels, not as Router inputs at inference. Full weights, distance definitions, temporal alignment, and confidence handling are given in Section A.4.

For a sampled tube $j$, adjacent checkpoints share the coarse latent, initial noise, and trajectory. Let $\hat{V}^{(r)}_{j,k}$ be the decoded output when that tube reaches checkpoint $k$ on replicate $r$, with $\hat{V}^{(r)}_{j,0} = \hat{V}^c$. The realized marginal gain and its regression target are

$$\Delta u^{(r)}_{j,k} = \mathcal{U}_{\mathrm{sem}}(\hat{V}^{(r)}_{j,k}) - \mathcal{U}_{\mathrm{sem}}(\hat{V}^{(r)}_{j,k-1}), \tag{9}$$

$$\bar{\Delta u}_{j,k} = \frac{1}{R} \sum_{r=1}^{R} \Delta u^{(r)}_{j,k}. \tag{10}$$

We suppress the common text and reference arguments. The target measures the return of extending this fixed refiner from $k-1$ to $k$; it can be negative when further refinement lowers utility. The corresponding value density on profiling hardware is

$$y_{j,k} = \mathrm{SVoC}(j,k) = \frac{\bar{\Delta u}_{j,k}}{\Delta c^{\mathrm{mean}}_{j,k}}, \qquad \Delta c^{\mathrm{mean}}_{j,k} = \mathcal{C}^{\mathrm{mean}}_{\mathrm{local}}(j,k;\mathcal{H}) - \mathcal{C}^{\mathrm{mean}}_{\mathrm{local}}(j,k-1;\mathcal{H}). \tag{11}$$

Sharing stochastic prefixes pairs the two utility measurements; averaging replicates estimates their expected difference. Label construction combines an initial stratified sample, value-guided sampling, uniform exploration, and successive halving. The full sampling procedure and variance identity are in Section A.6.

### 3.3 Marginal-Value Router

The Router observes only inference-time features:

$$e_j = [h^{\mathrm{latent}}_j, h^{\mathrm{text}}_j, a_j, \tau_j, c^{\mathrm{tube}}_j, c^{\mathrm{motion}}_j, c^{\mathrm{id}}_j], \tag{12}$$

including latent/text features, articulator and time identifiers, and confidence signals. A four-layer Transformer of width 384 predicts raw marginal gain for each candidate depth:

$$\widehat{\Delta u}_{j,k} = g_\phi(e_j, k). \tag{13}$$

All depth scores are predicted from the coarse state before refinement. Training combines raw-gain regression with a pairwise loss that orders gain per profiling-device cost; the complete objective is in Section A.1. At deployment,

$$v_{j,k} = \frac{\widehat{\Delta u}_{j,k}}{\widehat{\Delta c}^{\mathrm{mean}}_{j,k}}. \tag{14}$$

Cost normalization is applied to the allocation score, not to the raw regression target. Target-device calibration supplies the denominator and the complete-plan cost model; Table 6 reports the resulting deployment measurements.

### 3.4 Hardware-Budgeted Allocation and Execution

The cost model has two roles. An isolated-action mean-latency lookup table ranks increments through Eq. (14). A separately calibrated predictor estimates the P95 of the complete packed plan, including coarse generation, tube proposal, routing, packing, selected refinement, and decoding. Its features describe joint packed execution, so admission uses a plan estimate rather than a sum of isolated costs; definitions are in Section A.2.

For nominal residual allowance $r$, the configured total cap interpolates between the profiled coarse-only and unconditional-full P95 endpoints:

$$B^{\mathrm{total}}_r = C^{\mathrm{P95}}_{\mathrm{base}} + r\left(C^{\mathrm{P95}}_{\mathrm{full}} - C^{\mathrm{P95}}_{\mathrm{base}}\right). \tag{15}$$

Thus 50% refers to half of the profiled refinement allowance above the base cost, not half of full end-to-end latency. The cap, the predicted packed-plan P95 used for admission, and the final measured end-to-end P95 are distinct quantities. The detailed residual-budget notation and the RTX 4090 caps derived from the reported endpoints are given in Section B.

Only the next increment of each tube is available. A max-priority queue proposes the highest positive $v_{j,k}$ and accepts it if the updated packed plan satisfies $\widehat{\mathcal{C}}^{\mathrm{P95}}_{\mathrm{pack}} \le B^{\mathrm{total}}_r$. Acceptance unlocks the next depth of that tube; allocation ends when no queued positive-valued action remains. Queue operations require $O(JK\log(JK))$ time, in addition to the plan-cost evaluations. Algorithm 1 gives the pseudocode. The default scores remain fixed during selection; interaction-aware variants are analyzed separately in Section C.4.

Training first establishes the autoencoder, coarse generator, refiner, and semantic evaluator, then freezes them to construct labels, and finally trains the Router. The reported routing comparisons retain this fixed action space. Inference generates the complete coarse latent, proposes and scores tubes, allocates their depths, executes the packed refinements, and decodes the joint output. This is a non-streaming schedule with a single routing stage.

# 4 EXPERIMENTS

## 4.1 SETUP AND EVALUATION PROTOCOL

The Unified Controlled Evaluation (UCE) fixes test samples, the coarse generator, refiner, tube partition, held-out evaluators, metric implementations, seeds, target GPU, and P95 cap across routing policies. On PHOENIX14T and How2Sign Forster et al. (2014); Duarte et al. (2021), the primary metric is dataset-specific WER: Gloss WER against reference glosses and Transcript WER against normalized English transcripts, respectively. Comparisons are within datasets. We additionally report SiLVERScore, hand error, FVD, allocation regret, end-to-end P95, and budget exceedances. Unless stated otherwise, $r = 0.5$ in Eq. (15); results use three generator seeds crossed with five Router seeds, with "$\pm$" denoting standard deviations over the 15 combinations (Section B).

Default training labels use a frozen, dataset-specific in-house TwoStream-SLR-based evaluator A Chen et al. (2022). Independent B/C models are CorrNet Hu et al. (2023) and TwoStream-SLR recognizers on PHOENIX14T, and UniGloR Hwang et al. (2025) and FVLF Rao et al. (2026) video-to-English translators on How2Sign. Semantic metrics use SiLVERScore Imai et al. (2025) (D) and MediaPipe Holistic Grishchenko & Bazarevsky (2020) with an independent UniGloR pose encoder (E). A is used only for label construction; D is also used as an alternative label source in the sensitivity experiment of Table 4. All evaluators remain frozen across policies. Detailed roles and tokenization are in Section B.2; optimization and runtime settings are in Section F.2.

## 4.2 ROUTING UNDER THE LATENCY BUDGET

Table 1 compares routing policies on the same generator and refiner. Coarse-only and Full are quality/cost endpoints. The measured-value oracle supplies realized local marginal gains to the same greedy allocator, retaining its precedence and packed-plan rules. Its role is to isolate value prediction; the separate exact-allocation comparison is defined in Section C.4.

Table 1: PHOENIX14T UCE equal-latency routing at 50% residual budget. WER is Gloss WER; P95 is end-to-end latency.

| Policy | WER↓ | SiLVER↑ | Hand err.↓ | FVD↓ | P95 (ms)↓ |
|---|---|---|---|---|---|
| Coarse-only | 38.6 | .521 | 3.72 | 62.4 | 421 |
| Random | 30.8 | .582 | 3.06 | 58.1 | 676 |
| Uniform | 27.8 | .613 | 2.82 | 54.0 | 681 |
| Fixed Hand+Face ROI | 25.4 | .638 | 2.60 | 53.6 | 679 |
| Semantic-only | 24.1 | .651 | 2.51 | 53.0 | 682 |
| Predicted error reduction | 21.4 | .681 | 2.31 | 52.1 | 684 |
| Predicted cumulative gain (direct) | 21.0 | .686 | 2.26 | 51.9 | 683 |
| SEMAR | **20.1** | **.699** | **2.17** | **51.7** | 682 |
| Measured-value oracle | 18.4 | .721 | 1.98 | 50.8 | 680 |
| Full refinement | 17.3 | .726 | 1.91 | 50.3 | 1085 |

SEMAR obtains $20.1 \pm 0.38$ WER, compared with $21.4 \pm 0.45$ for predicted error reduction and $27.8 \pm 0.55$ for Uniform. At comparable 681–684 ms P95, its WER is lower by 7.7 points than Uniform, 1.3 points than predicted error reduction, and 0.9 points than predicted cumulative gain (direct). The measured-value oracle reaches 18.4 WER under the same allocation rule.

**Budget–quality trade-off.** Table 8 provides held-out WER at the nominal residual caps plotted in the left panel of Fig. 3. The right panel compares measured end-to-end latency at the 50% operating point with unconditional Full.

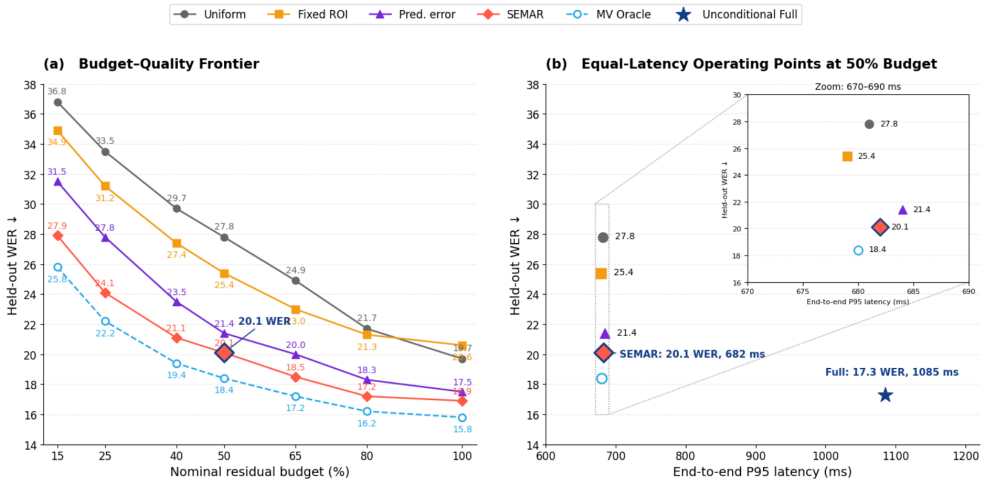



Figure 3: Verified UCE quality and latency operating points. Left: lower held-out WER is better. Right: P95 includes coarse generation, routing, packing, refinement, and decoding; Full is the unconditional 1085 ms endpoint. Lines connect measured points only.

At 50%, SEMAR reduces Full P95 from 1085 to 682 ms (37%, 1.59×), with a +2.8 WER gap. A 100% residual cap permits the full profiled allowance but does not force every action to execute: value-based skipping can produce a different output from unconditional Full. The complete nominal-budget WER points are reported in Section B.4.

### 4.3 Comparison with Contemporary Systems

Under the unified PHOENIX14T held-out protocol, SEMAR-50 obtains 20.1 Gloss WER at 682 ms end-to-end P95; the six contemporary configurations in Table 12 obtain 22.1–24.7 WER at 790–158000 ms. These are complete-system comparisons with different auxiliary resources and rendering paths, including a lexicon for SOKE full and predicted glosses for FGDM. The full results and reproduction settings are in Section E; the matched-backbone routing comparison is in Table 1.

### 4.4 Value Prediction and Supervision

Table 2: Compact value-prediction comparison on held-out atoms. Full ranking metrics are in Table 9.

| Predictor | $\rho_s \uparrow$ | Top-10 R↑ | Regret↓ |
|---|---|---|---|
| Semantic importance | .28 | 42.3 | 31.5% |
| Predicted error | .42 | 54.2 | 21.7% |
| Marginal gain/cost | **.63** | 71.0 | 10.5% |

On held-out atoms, marginal gain per cost yields $\rho_s = .63$ and 71.0 Top-10 recall, compared with .28/42.3 for semantic importance and .42/54.2 for the predicted-error proxy. Table 9 gives the complete ranking metrics. Value-oracle regret measures utility shortfall under the same allocator, rather than a WER difference or a gap to unconditional Full.

The qualitative allocation map in Fig. 4 shows selected hand events and supporting face context; quantitative comparisons are in Tables 1 and 2.

**Paired supervision and routing targets.** The shared-noise, nested-trajectory configuration reduces normalized marginal-label variance from 1.00 to .47. Table 10 compares this shared-prefix construction with an unpaired baseline under the same marginal-gain target. In Table 3, Cumulative gain (direct) denotes improvement over the coarse endpoint scored without differencing, Marginal gain denotes an adjacent-checkpoint improvement, and Marginal gain/cost denotes the cost-normalized

allocation score. WER and P95 use the UCE operating points in Table 1, while Pair Accuracy and value-oracle regret use the held-out-atom evaluation in Table 9; this source-specific convention avoids mixing metrics from different evaluation units. For the deployed Router, the regression head still predicts raw gain as in Eq. (13); cost enters its auxiliary ranking loss and inference score. Cumulative and marginal utility are related explicitly in Section A.4.

Table 3: Routing target ablation with identical Router architecture and data.

| Target | WER↓ | Pair Acc.↑ | Regret↓ | 8-step atoms↓ | P95↓ |
|---|---|---|---|---|---|
| Cumulative gain (direct) | 21.0 | 68.1 | 19.8% | 13.2% | 683 |
| Marginal gain | 20.7 | 73.0 | 12.8% | 9.8% | 683 |
| Marginal gain/cost | 20.1 | 75.0 | 10.5% | 7.9% | 682 |

Average NFE per selected tube is 5.6 for cumulative gain (direct), 4.9 for marginal gain, and 4.3 for marginal gain/cost. The corresponding 8-step atom fractions are 13.2%, 9.8%, and 7.9%. These depth summaries show that the marginal variants allocate less computation to deep refinement. On 500 exactly solvable small instances, greedy SVoC recovers 93.2% of exact utility with 6.8% median exact-allocation regret; the complete approximation and interaction analyses are in Section C.4.

### 4.5 Generalization and System Behavior

**Independent evaluators.** With evaluator A used for Router training, Uniform-to-SEMAR Gloss WER improves from 27.8 to 20.1 on held-out CorrNet (Recognizer B) and from 29.4 to 21.6 on held-out TwoStream-SLR (Recognizer C). SiLVERScore-D increases from .613 to .699 and UniGloR pose-semantic similarity-E from .648 to .726. Cross-evaluator ranking correlations are reported separately in Section D.2.

**Sensitivity to training utility.** Table 4 varies the evaluator-derived training utility while retaining the default evaluator-A utility as the reference. Label agreement is measured on the same held-out videos, tubes, candidate depth increments, stochastic trajectories, and replicate counts; only their utility scores change. Router models are retrained for each label source, and their selected actions are compared on matched test inputs and budgets. Removing either the sentence-similarity or gloss-CTC term changes the marginal-gain labels and selected refinement actions, while raising CorrNet Gloss WER by 0.6 or 0.9 points, respectively. Removing both terms produces a larger change: label-rank correlation falls to .59, action overlap to 57%, and CorrNet WER rises to 22.8. Using only the publicly available SiLVERScore Imai et al. (2025) to construct the utility gives .70 label-rank correlation, 68% action overlap, and 21.3 CorrNet WER. The TwoStream-SLR WER follows the same ordering across these settings. These point estimates support useful semantic signal beyond one in-house training utility, while the nonzero WER and allocation changes show that evaluator and utility choice still matter. SiLVERScore is an independent evaluation metric for the default A-trained Router, but not for the SiLVERScore-trained row; the B/C recognizers remain held out for every row. Metric definitions are in Section B.3.

Table 4: Training-utility sensitivity on PHOENIX14T. The three agreement columns compare with the complete evaluator-A utility; the first-row self-comparisons are definitional. Label $\rho_s$ is mean within-video Spearman correlation on raw marginal gains; sign agreement uses all common candidate increments. Action overlap is mean Jaccard agreement of selected tube–depth increments after separate Router training. WER uses held-out recognizers B/C, and P95 is end to end. SiLVERScore-only changes both evaluator and utility composition.

| Training label source | Label $\rho_s$ ↑ | Sign agree.↑ | Action overlap↑ | CorrNet WER↓ | TwoStream WER↓ | P95 (ms)↓ |
|---|---|---|---|---|---|---|
| Evaluator A, full utility | 1.00 | 100% | 100% | **20.1** | **21.6** | 682 |
| A, without sentence term | .82 | 87% | 77% | 20.7 | 22.2 | 680 |
| A, without gloss term | .76 | 83% | 71% | 21.0 | 22.5 | 679 |
| A, without both terms | .59 | 74% | 57% | 22.8 | 24.0 | 670 |
| SiLVERScore-only utility | .70 | 80% | 68% | 21.3 | 22.7 | 676 |

Table 5: Generalization under within-row UCE comparisons. PHOENIX14T rows use Gloss WER and How2Sign rows use Transcript WER; absolute values are not compared across datasets or backbone settings.

| Setting | Metric | Uniform-50 | Pred. error | SEMAR-50 | Rel. gain vs Uniform |
|---|---|---|---|---|---|
| PHOENIX14T / DGS | Gloss WER | 27.8 | 21.4 | **20.1** | 27.7% |
| How2Sign / ASL | Transcript WER | 34.5 | 29.0 | **26.8** | 22.3% |
| How2Sign signer-disjoint | Transcript WER | 31.6 | 24.9 | **22.6** | 28.5% |
| Refiner-B: conditional 3D U-Net | Gloss WER | 30.2 | 24.7 | **22.5** | 25.5% |

**Dataset and backbone comparisons.** Table 5 compares policies within each dataset/backbone setting. Refiner-B is an independently trained conditional 3D U-Net. For this refiner, marginal labels are reconstructed and the Router is retrained specifically for its action trajectories. The experiment therefore tests whether the routing protocol applies across refiner families, rather than zero-shot transfer of a Router trained for one refiner to another. The How2Sign rows are distinct from zero-shot DGS-to-ASL Router transfer, whose results and 5% adaptation analysis are in Section D.1.

Table 6: P95 latency and memory at 128 frames after per-GPU cost calibration. Exceed. is the packed-plan-level cap violation rate.

| GPU | Full | SEMAR-50 | Speedup | Overhead | Memory Full/SEMAR | Exceed. |
|---|---|---|---|---|---|---|
| RTX 4090 | 1085 | 682 | 1.59× | 6.7% | 15.2/10.9 GB | .8% |
| A100-80GB | 1180 | 755 | 1.56× | 6.1% | 18.6/13.2 GB | .6% |
| H100-80GB | 790 | 510 | 1.55× | 6.4% | 18.4/13.0 GB | .5% |

**End-to-end efficiency.** Per-device calibration yields 1.55–1.59× speedups at 128 frames. The active label-construction pipeline costs 94 A100-hours and 121 GB once, while the learned routing stage is used at inference. Additional clip lengths, calibration sample counts, and offline-cost comparisons are in Section F.1.

## 5 LIMITATIONS

Semantic supervision depends on the frozen evaluator, and the reported semantic evidence uses independent machine evaluators rather than a new human study. Local marginal gains approximate a joint utility that can contain hand–hand, hand–face, and temporal interactions. The fixed-score allocator also depends on useful preceding increments to reach deeper checkpoints. Routing cannot repair errors outside the fixed refiner's action space. The method requires offline labels, target-device cost calibration, and the complete coarse latent before allocation; zero-shot cross-language transfer remains weaker than in-language training.

## 6 CONCLUSION

SEMAR learns the semantic return of executable refinement increments and combines these predictions with measured hardware costs. Shared-prefix supervision links each prediction to an adjacent-depth utility difference, while packed-plan admission enforces the profiled latency cap. In matched-backbone evaluation, the method reaches 20.1 WER at 682 ms P95 and improves on importance-, error-, and total-gain-based routing at comparable latency. The results support marginal semantic utility as a useful target for budget-adaptive sign-language refinement.

## REPRODUCIBILITY STATEMENT

The main text specifies refinement actions, paired labels, Router scores, and budgeted execution. The appendix provides the full Router objective, allocator pseudocode, residual-budget and resampling

protocols, baseline adaptations, recorded hyperparameters, and hardware calibration. Table 13 summarizes the experimental settings recorded in this manuscript.

## ETHICS STATEMENT

Sign-language generation can change meaning through subtle errors in handshape, orientation, contact, or non-manual markers, so unverified deployment is inappropriate for high-stakes communication. This work conducts no new human-subject or signer/expert evaluation; reported experiments use existing datasets and machine-based evaluation protocols. Signer identity and video data require consent-aware handling.

## AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, and LaTeX revision. They were not used to create the experimental measurements reported in the paper. The authors remain responsible for the accuracy of the text, analyses, citations, and final submission.

## REFERENCES

Vasileios Baltatzis, Rolandos Alexandros Potamias, Evangelos Ververas, Guanxiong Sun, Jiankang Deng, and Stefanos Zafeiriou. Neural sign actors: A diffusion model for 3d sign language production from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1985–1995, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7793, 2018.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10023–10033, 2020.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. In *Advances in Neural Information Processing Systems*, 2022.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2sign: A large-scale multimodal dataset for continuous american sign language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2735–2744, 2021.

Sen Fang, Chunyu Sui, Yanghao Zhou, Xuedong Zhang, Hongbin Zhong, Minyu Zhao, Yapeng Tian, and Chen Chen. Signdiff: Diffusion models for american sign language production. *arXiv preprint arXiv:2308.16082*, 2023.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39, 2022.

Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *Language Resources and Evaluation Conference*, pp. 1911–1916, 2014.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Ivan Grishchenko and Valentin Bazarevsky. MediaPipe Holistic: Simultaneous face, hand and pose prediction, on device. Google Research Blog, December 2020. URL https://research.google/blog/mediapipe-holistic-simultaneous-face-hand-and-pose-prediction-on-device/.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Continuous sign language recognition with correlation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2529–2539, 2023.

Eui Jun Hwang, Sukmin Cho, Huije Lee, Youngwoo Yoon, and Jong C. Park. A spatio-temporal representation learning as an alternative to traditional glosses in sign language translation and production. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3352–3362, 2025.

Saki Imai, Mert Inan, Anthony B. Sicilia, and Malihe Alikhani. SiLVERScore: Semantically-aware embeddings for sign language generation evaluation. In *Proceedings of the 15th Conference on Recent Advances in Natural Language Processing*, pp. 452–461, 2025.

Zifan Jiang, Colin Leong, Amit Moryossef, Oliver Cory, Maksym Ivashechkin, Neha Tarigopula, Biao Zhang, Anne Göhring, Annette Rios, Rico Sennrich, and Sarah Ebling. Meaningful pose-based sign language evaluation. In *Proceedings of the Tenth Conference on Machine Translation*, pp. 64–80, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.wmt-1.4. URL https://aclanthology.org/2025.wmt-1.4/.

Tugce Kiziltepe, Sumeyye Meryem Tasyurek, and Hacer Yalim Keles. Iterative latent refinement for robust non-autoregressive sign language production. In *IEEE/CVF International Conference on Computer Vision Workshops*, pp. 4942–4952, 2025.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Jian Ma, Wenguan Wang, Yi Yang, and Feng Zheng. MS2SL: Multimodal spoken data-driven continuous sign language production. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7241–7254, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pp. 28100–28127, 2023.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Zhi Rao, Yucheng Zhou, Benjia Zhou, Yiqing Huang, Sergio Escalera, and Jun Wan. FVLF: A reinforcing vision-language framework for gloss-free sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9237–9247, June 2026.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjoern Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? In *Advances in Neural Information Processing Systems*, volume 34, pp. 2849–2861, 2021.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pp. 687–705. Springer, 2020.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Mixed signals: Sign language production via a mixture of motion primitives. In *IEEE/CVF International Conference on Computer Vision*, pp. 1919–1929, 2021.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5141–5151, 2022.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Tongkai Shi, Lianyu Hu, Fanhua Shang, Liqing Gao, and Wei Feng. Greg: Geometry-aware region refinement for sign language video generation. In *IEEE/CVF International Conference on Computer Vision*, pp. 16472–16481, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Sign language production using neural machine translation and generative adversarial networks. In *British Machine Vision Conference*, 2018.

Shengeng Tang, Jiayi He, Lechao Cheng, Jingjing Wu, Dan Guo, and Richang Hong. Discrete to continuous: Generating smooth transition poses from sign language observations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3481–3491, 2025.

Cong Wang, Zexuan Deng, Zhiwei Jiang, Yafeng Yin, Fei Shen, Zifeng Cheng, Shiping Ge, Shiwei Gan, and Qing Gu. Advanced sign language video generation with compressed and quantized multi-condition tokenization. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pp. 409–424, 2018.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8817–8826, 2018.

Pan Xie, Taiying Peng, Yao Du, and Qipeng Zhang. Sign language production with latent motion transformer. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3024–3034, 2024.

Maoxiao Ye, Xinfeng Ye, and Sathiamoorthy Manoharan. Hybrid autoregressive-diffusion model for real-time sign language production. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 750–763, 2026.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10809–10818, 2022.

Yiheng Yu, Sheng Liu, Yuan Feng, Zhelun Jin, Yining Jiang, and Min Xu. Focal-general diffusion model with semantic consistent guidance for sign language production. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35915–35925, 2026.

Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, 2020.

Ronglai Zuo, Rolandos Alexandros Potamias, Evangelos Ververas, Jiankang Deng, and Stefanos Zafeiriou. Signs as tokens: A retrieval-enhanced multilingual sign language generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23806–23816, October 2025.

# A Full Method and Allocation Details

## A.1 Full Router Objective

The Router is trained with direct regression and pairwise ranking. Let $\mathcal{D}$ denote all labeled increments and $\mathcal{P}$ the sampled ranking pairs. Then

$$\mathcal{L}_{\text{reg}} = \sum_{(j,k)\in\mathcal{D}} \text{Huber}(\bar{\Delta u}_{j,k} - \widehat{\Delta u}_{j,k}), \tag{16}$$

$$\mathcal{L}_{\text{rank}} = \sum_{(a,b)\in\mathcal{P}} \log\left(1 + \exp[-s_{ab}(\widehat{\Delta u}_a/\Delta c_a^{\text{mean}} - \widehat{\Delta u}_b/\Delta c_b^{\text{mean}})]\right), \tag{17}$$

$$\mathcal{L}_{\text{router}} = \mathcal{L}_{\text{reg}} + \lambda_r \mathcal{L}_{\text{rank}}, \tag{18}$$

where $s_{ab} = \text{sign}(y_a - y_b)$. The regression target is raw semantic gain, while the auxiliary ranking term uses profiling-device cost. The default model predicts all candidate depths from the coarse-state features; the iterative rescoring variant in Table 11 is evaluated separately.

## A.2 Full Measured-Cost Model

For hardware $\mathcal{H}$, the isolated-action lookup table is

$$\widehat{\mathcal{C}}_{\text{local}}^{\text{mean}}(j,k;\mathcal{H}) = \text{LUT}_{\mathcal{H}}^{\text{mean}}(N_j, T_j, H_jW_j, n_k), \tag{19}$$

where $N_j$, $T_j$, and $H_jW_j$ encode token count, temporal extent, and spatial area. The local incremental-cost proxy is

$$\widehat{\Delta c}_{j,k}^{\text{mean}} = \widehat{\mathcal{C}}_{\text{local}}^{\text{mean}}(j,k;\mathcal{H}) - \widehat{\mathcal{C}}_{\text{local}}^{\text{mean}}(j,k-1;\mathcal{H}). \tag{20}$$

This quantity normalizes the ranking score. Let $C_{\text{e2e}}(\mathcal{A};\mathcal{H})$ denote repeated end-to-end measurements and $f_{\text{pack}}(\mathcal{A})$ the packed-kernel features of a complete plan. Admission uses

$$\widehat{\mathcal{C}}_{\text{pack}}^{\text{P95}}(\mathcal{A};\mathcal{H}) = \widehat{Q}_{0.95}(C_{\text{e2e}}(\mathcal{A};\mathcal{H}) \mid f_{\text{pack}}(\mathcal{A})), \tag{21}$$

a conditional-quantile predictor fitted to complete-plan traces. A tentative selection is converted to the representation used by the execution kernel, whose joint token, extent, occupancy, and depth statistics form the predictor input. The local mean-cost table and the plan-level P95 predictor are calibrated separately.

## A.3 Allocator Pseudocode

## A.4 Action and Utility Conventions

An action $(j,k)$ extends one tube from $n_{k-1}$ to $n_k$ function evaluations. Selecting all three increments reaches 8 evaluations, not $2+4+8$. The mask in Eq. (2) specifies residual write-back, and Eq. (3)

**Algorithm 1** Budgeted semantic-value allocation

**Require:** coarse latent $z^c$, atoms $\{A_j\}_{j=1}^{J}$, text $X$, total latency budget $B = B_r^{\text{total}}$, hardware $\mathcal{H}$
Build inference states $e_j$ using Eq. (12)
Predict $\widehat{\Delta u}_{j,k}$ and compute $v_{j,k}$ for all $j, k$
Initialize max-priority queue $Q \leftarrow \{(j, 1) : j = 1, \ldots, J\}$ by Eq. (14)
$\mathcal{A} \leftarrow \emptyset$; $b \leftarrow B$
**while** $Q$ is not empty **do**
    $(j, k) \leftarrow \textsc{PopMax}(Q)$
    **if** $v_{j,k} \leq 0$ **then**
        **break** ▷ all remaining queued values are non-positive
    **end if**
    $\mathcal{A}' \leftarrow \mathcal{A} \cup \{(j, k)\}$
    $c' \leftarrow \widehat{\mathcal{C}}_{\text{pack}}^{\text{P95}}(\mathcal{A}'; \mathcal{H})$
    **if** $c' > B$ **then**
        **continue**
    **end if**
    $\mathcal{A} \leftarrow \mathcal{A}'$; $b \leftarrow B - c'$
    **if** $k < K$ **then**
        $\textsc{Push}(Q, (j, k + 1))$
    **end if**
**end while**
Pack selected atoms and execute their deepest selected checkpoints
**return** decoded video $\hat{V}_{\mathcal{A}}$, selection $\mathcal{A}$, unused budget $b$

specifies how overlapping selected tubes are combined. The complete-plan timing accounts for the packed execution associated with these actions.

The frozen evaluator exposes

$$E_{\text{train}}(V) = \{e_{\text{sent}}, p_{\text{gloss}}, h_{\text{hand}}, h_{\text{orient}}, h_{\text{contact}}, h_{\text{face}}, h_{\text{motion}}\}. \tag{22}$$

In Eq. (8), the fixed weights are $(.30, .25, .18, .10, .10, .07)$ for sentence similarity, gloss CTC, handshape, orientation, contact, and non-manual discrepancy, respectively. Each component is robustly normalized using its validation-split median and interquartile range; these statistics and the weights are frozen before marginal-label construction.

Sentence similarity is cosine similarity between frozen video and text embeddings, and gloss discrepancy is length-normalized CTC loss. Handshape uses confidence-weighted cosine distance between frozen per-hand features. Orientation uses the geodesic discrepancy between estimated palm coordinate frames, or between the corresponding palm-normal and wrist-to-middle-finger directions when only keypoints are available. Contact is confidence-weighted multi-label binary cross-entropy over hand–hand, hand–face, and hand–torso events. The non-manual term combines frozen facial-feature distance with facial-action discrepancies.

A shared monotonic DTW path aligns the local terms using body and hand motion features within a 10% Sakoe–Chiba window; sentence similarity and CTC remain unaligned. Frames below .5 local-label confidence are excluded. For each example, missing supervised terms are omitted rather than assigned an observed zero error, and the active weights are renormalized to sum to one. PHOENIX14T supplies gloss annotations; How2Sign omits the gloss term unless a verified gloss sequence is available.

For the same stochastic replicate, define cumulative gain over the coarse endpoint by

$$G_{j,k}^{(r)} = \mathcal{U}_{\text{sem}}(\hat{V}_{j,k}^{(r)}) - \mathcal{U}_{\text{sem}}(\hat{V}^c) = \sum_{\ell=1}^{k} \Delta u_{j,\ell}^{(r)}. \tag{23}$$

This identity distinguishes cumulative improvement from the return of the next increment. The cost-normalized marginal score adds a third quantity, $v_{j,k}$ in Eq. (14). These definitions distinguish the target and scoring quantities named in Table 3.

### A.5 MATCHED-BACKBONE ROUTING CONTROLS

All routing policies share the coarse generator, residual refiner, tube proposals, candidate increments, 2/4/8-evaluation precedence constraints, test samples and seeds, target-device cost lookup table, packed-plan P95 predictor, total admission cap, masked write-back, and decoder. They differ only in the priority assigned to the next available increment and, for learned policies, its supervision target. For action $a = (j, k)$, define

$$\widehat{\Delta c}_{j,k} = \widehat{C}_{\text{local}}(j, k) - \widehat{C}_{\text{local}}(j, k-1). \tag{24}$$

Every policy uses the same admission test $\widehat{C}^{\text{P95}}_{\text{pack}}(\mathcal{A} \cup \{a\}) \le B^{\text{total}}_r$; an accepted action alone unlocks the next depth of its tube.

**Non-learned controls.** For every sample–seed plan, Random draws one fixed $s_a \sim \text{Uniform}(0, 1)$ per action and preserves that ordering throughout allocation; results average five routing seeds. Uniform uses depth-wise round-robin allocation across tubes, considering every feasible first increment before any second increment and every feasible second increment before any third. Within-depth ties use a fixed content-independent hash, and infeasible actions are skipped. Fixed Hand+Face ROI sets $q_j = 1$ for left-hand, right-hand, and face tubes and zero otherwise, and scores eligible actions as

$$s^{\text{ROI}}_{j,k} = q_j / \widehat{\Delta c}_{j,k}, \tag{25}$$

with the same tie-breaking rule; its weights are not adapted to test content.

**Learned controls.** Semantic-only predicts the depth-independent masking-sensitivity target

$$I_j = \mathcal{U}_{\text{sem}}(V) - \mathcal{U}_{\text{sem}}(\text{Mask}(V, j)), \qquad s^{\text{sem}}_{j,k} = \max(0, \widehat{I}_j) / \widehat{\Delta c}_{j,k}. \tag{26}$$

Predicted error reduction excludes sentence similarity and gloss CTC and uses

$$D_{\text{local}} = w_h D_{\text{hand}} + w_o D_{\text{orient}} + w_c D_{\text{contact}} + w_f D_{\text{nmm}}, \tag{27}$$

$$\Delta e_{j,k} = D_{\text{local}}(\hat{V}_{j,k-1}, V) - D_{\text{local}}(\hat{V}_{j,k}, V), \qquad s^{\text{err}}_{j,k} = \widehat{\Delta e}_{j,k} / \widehat{\Delta c}_{j,k}, \tag{28}$$

and admits only positive predicted reductions. Predicted cumulative gain (direct) learns $G_{j,k}$ from Eq. (23) and uses $s^{\text{cum}}_{j,k} = \widehat{G}_{j,k} / \widehat{\Delta c}_{j,k}$ directly. We did not evaluate a separate cumulative-predict-then-difference variant. SEMAR instead uses $s_{j,k} = \widehat{\Delta u}_{j,k} / \widehat{\Delta c}_{j,k}$. The measured-value oracle replaces the prediction with $\overline{\Delta u}_{j,k}$ while retaining the same greedy queue, precedence, and packed-plan admission; it is not the exact-allocation oracle of Section C.4.

Semantic-only, predicted error reduction, predicted cumulative gain (direct), and SEMAR use the same Router architecture, trajectory cache, number of supervised actions, train/validation split, optimizer, batch size, training steps, generator and Router seeds, and checkpoint-selection rule. Each target is normalized using its own validation median and interquartile range, frozen before training. All use standard Huber regression with $\delta = 1$ plus the pairwise ranking loss in Eq. (18); $\lambda_r$ is selected once on validation data and then held fixed across the learned controls.

### A.6 PAIRED-LABEL CONSTRUCTION

Active labeling begins with a score-free seed round that consumes 10% of the trajectory budget. Sampling is stratified by dataset, articulator class, temporal position (beginning, middle, or end), and refinement depth. Low-frequency contact, occlusion, and low-confidence-tube strata receive minimum quotas. Executed atoms form the seed cache for a warm-start Router. Four acquisition rounds then select new atoms with a fixed mixture: 50% are the unlabeled atoms with highest predicted SVoC, 25% are historical hard negatives (predicted high-value atoms whose observed gain is non-positive), and 25% are sampled uniformly from the remaining strata.

Every acquired atom first receives two shared-prefix trajectory replicates, exposing the applicable adjacent pairs among the nested 0/2/4/8 checkpoints. Within each articulator–depth stratum, atoms are ranked by

$$a_{j,k} = \overline{\Delta u}_{j,k} - \text{SE}(\Delta u_{j,k}). \tag{29}$$

The top half receive four replicates, after which the top half receive eight. For a stratum with $n$ candidates, an advancement stage retains $\min\{n, \max(8, \lceil n/2 \rceil)\}$ atoms, so a small stratum is not eliminated prematurely. Non-advanced atoms retain their current replicate means. Thus successive halving increases the number of stochastic replicates for a fixed atom–checkpoint label; it does not redefine or skip the nested checkpoint depths.

Before acquisition scores are applied in each round, 5% of candidate atoms are uniformly assigned to a held-out audit subset with coverage across articulators and depths. Audit atoms do not participate in halving: each is evaluated with eight replicates at every 0/2/4/8 checkpoint. The audit records early-stopping error, gain-sign agreement, missed-positive rate, top-decile value recall, Router-ranking correlation, and the downstream difference between Routers trained with halved and exhaustive labels; no unmeasured audit outcome is reported here. Five master random seeds are fixed in advance and determine stratified sampling, audit assignment, acquisition tie-breaking, and trajectory noise.

Cache entries are keyed by generator and refiner checkpoints, evaluator version, local-cost table version, and packed-cost predictor version, and are invalidated when any of these components changes. The reported 23% trajectory-execution rate uses as its denominator all eligible atom–checkpoint–replicate executions in the exhaustive counterfactual labeling protocol. It is an execution count rather than a derivation of storage or GPU time; the measured offline totals are 94 A100-hours and 121 GB, as detailed in Section F.1.

For two adjacent utility measurements $U_k$ and $U_{k-1}$,

$$\mathrm{Var}(U_k - U_{k-1}) = \mathrm{Var}(U_k) + \mathrm{Var}(U_{k-1}) - 2\,\mathrm{Cov}(U_k, U_{k-1}). \tag{30}$$

Shared noise and prefixes couple the measurements so that common stochastic variation can cancel in the difference. This explains the variance-reduction mechanism; the reported empirical comparisons are those in Table 10. Replicate averaging estimates the expected gain for the fixed action and coarse state.

# B Full Evaluation Protocol

## B.1 Residual Budgets and Statistical Aggregation

A nominal residual budget $r \in [0, 1]$ scales the refinement allowance above no refinement. For each target device, the coarse-only and unconditional-full endpoints are profiled with the same end-to-end timing protocol. Define

$$C^{\mathrm{P95}}_{\mathrm{base}}(\mathcal{H}) = \mathcal{C}^{\mathrm{P95}}_{\mathrm{e2e}}(\emptyset; \mathcal{H}), \qquad C^{\mathrm{P95}}_{\mathrm{full}}(\mathcal{H}) = \mathcal{C}^{\mathrm{P95}}_{\mathrm{e2e}}(\mathcal{A}_{\mathrm{full}}; \mathcal{H}), \tag{31}$$

where $\mathcal{A}_{\mathrm{full}}$ executes every eligible increment. Then

$$B^{\mathrm{res}}_{100} = C^{\mathrm{P95}}_{\mathrm{full}} - C^{\mathrm{P95}}_{\mathrm{base}}, \qquad B^{\mathrm{res}}_{r} = r B^{\mathrm{res}}_{100}, \qquad B^{\mathrm{total}}_{r} = C^{\mathrm{P95}}_{\mathrm{base}}(\mathcal{H}) + B^{\mathrm{res}}_{r}. \tag{32}$$

This is equivalent to Eq. (15). On the RTX 4090, the reported measured endpoints are $C^{\mathrm{P95}}_{\mathrm{base}} = 421$ ms and $C^{\mathrm{P95}}_{\mathrm{full}} = 1085$ ms, giving a residual allowance of 664 ms. The resulting caps in Table 7 are derived from these endpoints and rounded to the nearest millisecond; they are not direct timing measurements.

Table 7: RTX 4090 residual allowances and total P95 admission caps derived from the reported coarse-only and unconditional-full endpoints.

| Residual budget | Residual allowance (ms) | Total P95 cap (ms) |
|---:|---:|---:|
| 15% | 100 | 521 |
| 25% | 166 | 587 |
| 40% | 266 | 687 |
| 50% | 332 | 753 |
| 65% | 432 | 853 |
| 80% | 531 | 952 |
| 100% | 664 | 1085 |

At 50%, SEMAR has a measured end-to-end P95 of 682 ms. It therefore uses $(682 - 421)/332 = 78.6\%$ of the available residual allowance, or $682/753 = 90.6\%$ of the total cap. Unused allowance can arise from non-positive remaining values, discrete refinement depths, or the inability to admit another packed plan without exceeding the cap. A 100% cap therefore permits, but does not require, unconditional Full execution. P95 includes coarse generation, tube proposal, routing, packing, selected refinement, and decoding. Because the coarse-only P95 endpoints for A100 and H100 are not reported, we do not derive numerical caps or utilization values for those devices.

Budget exceedance is evaluated at the packed-plan level. Each selected sample–generator-seed–Router-seed plan is executed 100 times with synchronization; the first 10 executions are discarded as warm-up, and empirical P95 is computed over the remainder. A plan is an exceedance when its measured empirical P95 is greater than its configured total cap:

$$\text{ExceedRate} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[C^{\text{P95}}_{\text{measured},i} > B^{\text{total}}_{r,i}\right], \tag{33}$$

where $N$ is the number of evaluated sample–seed plans, not the number of individual timing repetitions. Thus the reported RTX 4090 rate of .8% means that .8% of final packed plans exceed their configured caps; it is distinct from the fraction of individual repetitions above a P95 threshold.

Main results aggregate three generator seeds and five Router seeds (15 combinations); "±" denotes standard deviation across those combinations, not a confidence interval for a policy difference. Seed combinations are repeated stochastic measurements rather than independent sentence observations. Because paired sentence-level bootstrap outputs are not reported, we make no significance claims from the seed-level deviations.

### B.2 Evaluator Roles and Metric Scope

Evaluator A is a frozen in-house multi-task sign-perception model built on the RGB/keypoint two-stream design of TwoStream-SLR Chen et al. (2022). Separate PHOENIX14T and How2Sign instances are trained on their respective training splits. Its pooled representation feeds a video–text alignment head, its temporal representation feeds a gloss CTC head when glosses are available, and separate local heads provide handshape, orientation, contact, and non-manual features. Evaluator A is used only to construct training utility and marginal labels.

For PHOENIX14T, Recognizer B is CorrNet Hu et al. (2023) and Recognizer C is an independently initialized and trained TwoStream-SLR model. Both decode gloss sequences with CTC and are frozen for evaluation. For How2Sign, B and C are independently trained UniGloR Hwang et al. (2025) and FVLF Rao et al. (2026) video-to-English translators. They produce English text rather than pseudo-glosses. Evaluator D is the official SiLVERScore implementation Imai et al. (2025). Evaluator E applies MediaPipe Holistic Grishchenko & Bazarevsky (2020) to extract body, hand, and face landmarks and uses an independently trained UniGloR temporal pose encoder; its reported metric is cosine pose-semantic similarity. B/C do not share checkpoints with A or participate in marginal-label construction or Router optimization, and are not jointly trained with the generator or Router. D/E do not share A's prediction heads, and all evaluators are frozen. D serves as a held-out metric for the default A-trained Router, but its score constructs labels in the SiLVERScore-only sensitivity row of Table 4; B/C remain held out there.

For PHOENIX14T, Gloss WER is computed against the reference gloss sequence. For How2Sign, which lacks an equivalent benchmark-wide gloss target, Transcript WER is computed against the English reference transcript. Reference and prediction strings undergo Unicode normalization, lowercasing, punctuation removal, contraction and whitespace normalization, conversion of numbers to a common textual form, and whitespace tokenization. We then compute standard word-level Levenshtein error,

$$\text{WER} = \frac{S + D + I}{N}, \tag{34}$$

where $S$, $D$, and $I$ are substitutions, deletions, and insertions and $N$ is the number of reference tokens. In each UCE comparison, policies share the evaluation sample set, reference targets, preprocessing, and metric implementation. Gloss WER and Transcript WER are interpreted only within their respective datasets.

### B.3 Training-Utility Sensitivity Protocol

The label-agreement comparison uses a common held-out set of videos with identical tube proposals, candidate increments $a = (j, k)$ at $0 \to 2$, $2 \to 4$, and $4 \to 8$ network evaluations, stochastic trajectories, and replicate counts. The fixed checkpoint videos are rescored under each utility; this isolates the effect of the evaluator and utility definition on label construction. For ablations of evaluator A, omitted terms are removed and the remaining active weights are renormalized as in Section A.4. The SiLVERScore-only condition instead uses its video–text score as the utility, without the other five evaluator-A components. Let $\mathcal{A}_x$ be the common candidate-increment set for video $x$, and let $\bar{\Delta u}^{A}_{x,a}$ and $\bar{\Delta u}^{E}_{x,a}$ denote replicate-mean marginal gains from the complete A utility and a comparison utility $E$.

The reported label-rank statistic first computes Spearman correlation across the *within-video* candidate increments and then averages across videos for which the correlation is defined:

$$\rho_{\text{label}} = \frac{1}{|\mathcal{X}_\rho|} \sum_{x \in \mathcal{X}_\rho} \text{Spearman}\Big( \{\bar{\Delta u}^{A}_{x,a}\}_{a \in \mathcal{A}_x}, \{\bar{\Delta u}^{E}_{x,a}\}_{a \in \mathcal{A}_x} \Big), \tag{35}$$

where $\mathcal{X}_\rho$ excludes videos with an undefined correlation, such as a constant label vector. The sign-agreement column compares all common candidates, including near-zero gains:

$$\text{SignAgree} = \frac{\sum_x \sum_{a \in \mathcal{A}_x} \mathbf{1}\Big[\text{sign}(\bar{\Delta u}^{A}_{x,a}) = \text{sign}(\bar{\Delta u}^{E}_{x,a})\Big]}{\sum_x |\mathcal{A}_x|}. \tag{36}$$

Near-zero sign differences may reflect replicate noise, so this statistic is interpreted together with rank agreement and downstream routing rather than as a noise-free semantic judgment. Because utility scales differ, no shared uncalibrated numerical threshold is applied to define a near-zero gain.

For action overlap, a separate Router is trained for each label source and run on matched test examples, stochastic seeds, latency budgets, and cost models. Let $S^A_x$ and $S^E_x$ contain the selected executable $(j, k)$ increments, including their depth, for the two Routers. The table reports

$$\text{ActionJaccard} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{|S^A_x \cap S^E_x|}{|S^A_x \cup S^E_x|}, \tag{37}$$

with a both-empty pair assigned agreement one. Label correlation and sign agreement therefore compare *fixed-video training targets*, whereas action overlap compares *trained routing plans*. The two comparisons answer different questions: similar label ranks need not imply identical selected plans near a latency boundary.

### B.4 Full Budget–Quality Operating Points

Table 8: UCE held-out WER over nominal residual budgets. The 100% cap still permits value-based skipping and is distinct from unconditional Full. MV oracle denotes the measured-value oracle defined in Table 1.

| Budget | Uniform | Fixed ROI | Pred. error | SEMAR | MV oracle |
|---|---|---|---|---|---|
| 15% | 36.8 | 34.9 | 31.5 | **27.9** | 25.8 |
| 25% | 33.5 | 31.2 | 27.8 | **24.1** | 22.2 |
| 40% | 29.7 | 27.4 | 23.5 | **21.1** | 19.4 |
| 50% | 27.8 | 25.4 | 21.4 | **20.1** | 18.4 |
| 65% | 24.9 | 23.0 | 20.0 | **18.5** | 17.2 |
| 80% | 21.7 | 21.3 | 18.3 | **17.2** | 16.2 |
| 100% cap | 19.7 | 20.6 | 17.5 | **16.9** | 15.8 |

Table 8 tabulates WER at nominal residual caps; the equal-latency operating point and its measured end-to-end latency are reported in Table 1.

# C Additional Ablations and Allocator Analysis

## C.1 Full Value-Prediction Metrics

Table 9: Full value prediction on held-out atoms. Value-oracle regret is the normalized utility shortfall relative to the measured-value oracle under the same greedy allocator and packed-plan budget.

| Predictor | $\rho_s \uparrow$ | Pair Acc.↑ | Top-10 R↑ | NDCG↑ | Value regret↓ |
|---|---|---|---|---|---|
| Semantic importance | .28 | 61.2 | 42.3 | .49 | 31.5% |
| Predicted error | .42 | 67.0 | 54.2 | .62 | 21.7% |
| Predicted cumulative gain (direct) | .46 | 68.1 | 57.2 | .65 | 19.8% |
| Marginal gain | .58 | 73.0 | 67.5 | .75 | 12.8% |
| Marginal gain/cost | **.63** | **75.0** | **71.0** | **.80** | **10.5%** |

The measured-value oracle replaces predicted scores with realized local marginal values while retaining the same greedy queue, precedence constraints, local cost normalization, and packed-plan admission. Value-oracle regret measures the normalized utility shortfall relative to this reference. It is distinct from the exact-allocation regret in Section C.4, which uses jointly evaluated action sets.

## C.2 Shared-Prefix Comparison

Table 10: Shared-prefix supervision compared with an unpaired baseline using the same marginal-gain target.

| Supervision | Label Var.↓ | SNR↑ | Pair Acc.↑ | WER↓ |
|---|---|---|---|---|
| Unpaired noise and trajectories | 1.00 | 1.00 | 64.5 | 22.9 |
| Shared-noise nested trajectory | **.47** | **1.93** | **72.1** | **20.7** |

Both rows use a marginal-gain target. The shared-prefix row uses the common-noise, nested-checkpoint trajectory described in Section 3.2; the unpaired baseline draws independent noise and executes independent trajectories for adjacent checkpoints. Shared-prefix supervision reduces normalized label variance from 1.00 to .47, raises SNR from 1.00 to 1.93 and pair accuracy from 64.5 to 72.1, and improves WER from 22.9 to 20.7. The latter is the marginal-gain supervision result, distinct from the 20.1 WER of the cost-normalized deployed policy.

### C.3 Qualitative Allocation

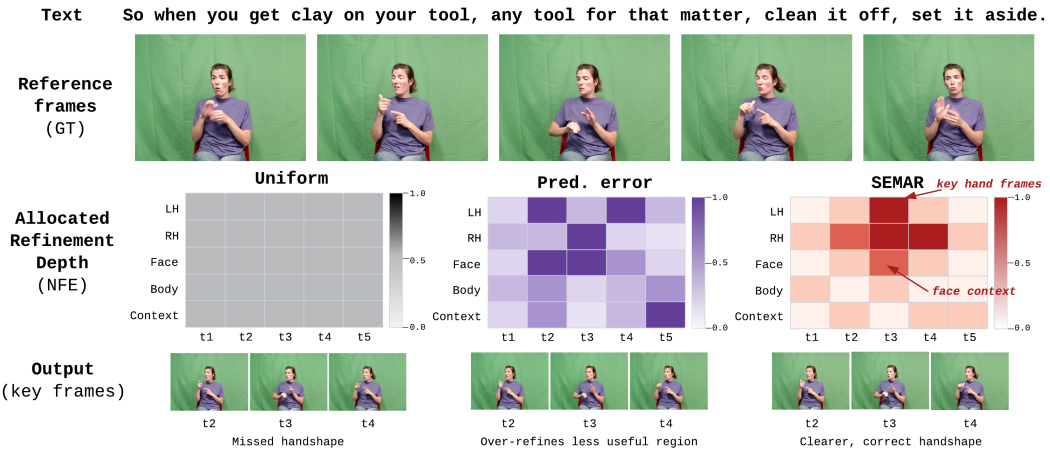



Figure 4: Qualitative comparison of spatiotemporal refinement allocation for a representative sign sequence. From left to right, uniform routing distributes computation evenly, predicted-error routing concentrates on visually uncertain regions, and SEMAR prioritizes semantically valuable articulator–time locations, including key hand frames and supporting face context. Brighter cells indicate greater allocated refinement compute; the bottom rows show representative frames selected by each policy.

### C.4 Exact Allocation and Interaction Stress

On 500 exactly solvable instances with 4–8 tubes and at most three increments per tube, the exact-allocation oracle enumerates all precedence- and budget-feasible action sets and evaluates each through joint refinement and final-video semantic utility. Greedy SVoC has 6.8% median and 14.5% P90 exact-allocation regret, recovering 93.2% of exact utility. Pair-aware features reduce these to 5.2% and 11.8%, raising recovered utility to 94.8%; greedy predicted-error allocation recovers 81.5%.

The default Router estimates local gains at the fixed coarse state. The corresponding conditional joint gain would instead be

$$\Delta U(a \mid \mathcal{A}) = \mathcal{U}_{\text{sem}}(\hat{V}_{\mathcal{A}\cup\{a\}}) - \mathcal{U}_{\text{sem}}(\hat{V}_{\mathcal{A}}), \tag{38}$$

which can change as other tubes are selected. Overlap normalization in Eq. (3), two-hand contact, and hand–face contact couple action effects. A separate depth effect arises when a low- or negative-value prefix precedes a valuable deeper checkpoint: the one-step queue only unlocks a depth after accepting its predecessor. The exact-allocation comparison evaluates the complete feasible action set rather than assuming either local additivity or nonnegative gains.

Table 11: Normalized exact-allocation regret under controlled interaction stress.

| Interaction | Static SVoC | + Context | + Pair | Iterative rescore |
|---|---|---|---|---|
| Low | 5.5% | 5.0% | 4.8% | **4.2%** |
| Medium | 10.2% | 8.9% | 8.0% | **6.7%** |
| High | 19.5% | 17.0% | 14.8% | **11.2%** |

Static SVoC regret increases with interaction intensity, reaching 19.5% in the high-interaction setting. Low-resolution context, pair-aware features, and iterative rescoring reduce this gap, with iterative rescoring reaching 11.2%. The default SEMAR policy uses low-resolution context and one fixed prediction pass; pair-aware and rescoring variants are confined to this analysis.

# D GENERALIZATION AND TRANSFER

## D.1 CROSS-LANGUAGE ROUTER TRANSFER

A DGS Router obtains 29.8 Transcript WER zero-shot on How2Sign. Adapting only the Router with a 5% target-domain label cache improves Transcript WER to 27.4. Relative to the in-language How2Sign result of 26.8, this recovers $(29.8 - 27.4)/(29.8 - 26.8) = 80.0\%$ of the transfer gap. This experiment tests transfer of a trained Router, whereas Table 5 compares routing policies within dataset/backbone settings. In-language training remains the stronger result.

## D.2 CROSS-EVALUATOR AGREEMENT

For the default evaluator-A-trained Router, independent Recognizers B/C and semantic Evaluators D/E preserve the reported policy ordering. Policy-ranking correlations are .78 between CorrNet and TwoStream-SLR and .73 between CorrNet and SiLVERScore.

# E EXTERNAL BASELINE REPRODUCTION DETAILS

Table 12: Contemporary SLP/SLVG systems under the unified held-out protocol on PHOENIX14T. WER is Gloss WER; "Pred. gloss" denotes predicted gloss input. These are complete-system comparisons with the listed resource differences.

| Method | System type | Extra resource | WER↓ | SiLVER↑ | Hand err.↓ | P95 (ms)↓ |
|---|---|---|---|---|---|---|
| SignViP Wang et al. (2025) | Text→Video | No | 22.6 | .682 | 2.22 | 158000 |
| SOKE w/o retr. Zuo et al. (2025) | Text→3D Pose→Video | No | 23.5 | .669 | 2.31 | 1220 |
| SOKE full Zuo et al. (2025) | Text→3D Pose→Video | Lexicon | 22.1 | .686 | 2.19 | 1300 |
| GReg-adapted Shi et al. (2025) | Regional refine | No | 23.4 | .661 | 2.37 | 790 |
| HybridSign Ye et al. (2026) | Text→Pose→Video | No | 24.6 | .658 | 2.39 | 18600 |
| FGDM-PredGloss Yu et al. (2026) | Text→Gloss→Pose→Video | Pred. gloss | 24.7 | .656 | 2.31 | 1350 |
| **SEMAR-50** | Text→Video + adaptive refine | No | **20.1** | **.699** | **2.17** | **682** |

**SignViP and SOKE.** SignViP Wang et al. (2025) is evaluated with its 50-step diffusion inference configuration. SOKE Zuo et al. (2025) is evaluated without retrieval and in its full lexicon-assisted setting; the lexicon is listed as an extra resource in Table 12. Both SOKE pose outputs use the common renderer.

**Regional and hybrid refinement.** GReg Shi et al. (2025) is connected to the common coarse motion and decoder, yielding the GReg-adapted configuration. HybridSign Ye et al. (2026) produces poses that are passed through the common renderer. Table 12 reports complete-sequence P95; HybridSign's streaming design additionally concerns time to first frame, a different latency quantity.

**Gloss-conditioned generation.** FGDM Yu et al. (2026) is natively gloss-to-pose. Its predicted-gloss setting supplies the text-input baseline in Table 12; ground-truth-gloss conditioning is a separate resource setting.

All numbers in Table 12 use the unified held-out protocol. Original-paper BLEU, ROUGE, WER, DTW, FID/FVD, SSIM, LPIPS, and region-error values serve as provenance checks rather than numerically interchangeable UCE results.

# F SYSTEM AND REPRODUCIBILITY DETAILS

## F.1 ADDITIONAL SYSTEM AND OFFLINE-COST ANALYSES

At 64 frames, speedups are 1.51–1.55×; at 256 frames, they are 1.59–1.62×. The 128-frame per-GPU measurements, memory use, routing overhead, and exceedance rates are in Table 6.

Transferring an A100 cost table directly to an RTX 4090 produces 4.6% budget exceedances. Using 50, 100, 250, 500, and 1000 target-device calibration traces reduces this to 2.8, 1.9, 1.2, .8, and .6%,

respectively. The reported configuration uses 500 traces and therefore reports .8% exceedances in Table 6; doubling the calibration count reduces the rate by .2 percentage points.

Exhaustive counterfactual labeling requires 410 A100-hours and 520 GB. Value-stratified sampling with successive halving and caching executes 23% of trajectories, costs 94 A100-hours and 121 GB, and obtains 20.1 WER with 10.5% value-oracle regret. The GPU-hour reduction is approximately 77%. Label construction is a one-time offline cost, separate from the end-to-end inference latency in Tables 1 and 6.

### F.2 Recorded Experimental Configuration

Table 13 gives the reported video, tube, Router, optimization, seed, and hardware settings. Videos are sampled at 25 fps, with $4\times$ temporal and $8\times$ spatial autoencoder compression. Clips contain 64–128 frames; temporal atoms cover 16 input frames with a four-frame overlap. The Router has four Transformer layers of width 384 and one marginal-gain head, trained with AdamW, peak learning rate $2 \times 10^{-4}$, cosine decay, and 64 atoms per GPU. Label-cache experiments use five seeds at the 10% cache operating point and one seed elsewhere; the main routing results use the 15 generator–Router combinations described in Section B.

Table 13: Reproducibility configuration recorded for the reported experiments.

| Item | Recorded setting |
|---|---|
| Input/video latent | 25 fps; 64–128 frames; temporal compression 4; spatial compression 8 |
| Decision atoms | 16 input frames; four-frame overlap; hand, face, body, and context tubes |
| Residual trajectory | Shared initial noise; nested checkpoints at 0/2/4/8 function evaluations |
| Default refiner | Latent-video Diffusion Transformer; 12 Transformer blocks; cross-attention text/gloss conditioning; adaptive-layer-normalization timestep/signer conditioning |
| Refiner-family control | Independently trained conditional 3D U-Net; labels reconstructed and Router retrained |
| Semantic utility | Weights $(.30, .25, .18, .10, .10, .07)$; validation median/IQR normalization; .5 local-confidence threshold |
| Local alignment | Shared monotonic DTW path; body/hand motion features; 10% Sakoe–Chiba window |
| Router | Four Transformer layers; width 384; one marginal-gain head |
| Router optimization | AdamW; peak LR $2 \times 10^{-4}$; cosine decay; 64 atoms/GPU |
| Result seeds | Three generator $\times$ five Router seeds (15 combinations) |
| Label construction | Active sampling + successive halving; 23% trajectories; 94 A100-h; 121 GB |
| Latency calibration | Per-GPU calibration; 500 traces; batch 1; end-to-end P95 including decode |
| Budget instantiation | RTX 4090 measured endpoints 421/1085 ms; derived 50% total cap 753 ms |
| Exceedance timing | 100 synchronized executions/plan; first 10 discarded; empirical P95 over remaining runs |
| Evaluation models | A: in-house TwoStream-based multi-task evaluator; B/C: CorrNet and TwoStream-SLR on PHOENIX14T, UniGloR and FVLF on How2Sign; D: SiLVERScore; E: MediaPipe Holistic + UniGloR pose encoder |
| Metric targets | PHOENIX14T reference glosses (Gloss WER); How2Sign normalized English transcripts (Transcript WER) |
| Target hardware | RTX 4090, A100-80GB, and H100-80GB |

Local timing uses repeated isolated actions indexed by token count, temporal extent, spatial area, and depth. Plan-level calibration uses the complete execution path described in Section A.2. Cache versions jointly identify the generator, evaluator, local-cost table, and packed-plan predictor. Changes to these dependencies invalidate cached labels or cost associations.