JFER: JOINT FUTURE EXPLORATION AND REASONING FOR INTERACTIVE MOTION FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive motion forecasting provides downstream planning with a small set of joint futures, although the number of possible agent-intention combinations can far exceed the prediction budget. Similar hypotheses may occupy several output slots, leaving other plausible interactions unrepresented. We introduce JFER, a Joint Future Exploration and Reasoning framework that addresses this limitation through structured hypothesis construction and complementary exploration. JFER organizes agent-level intentions into a joint-intention bank and assigns six prediction slots to selected pairs before full trajectory decoding. It then identifies additional intentions using their predicted support and distance from the decoded joint endpoints. Each selected intention guides the adaptation of an existing trajectory pair, producing a complementary hypothesis without another full decoding pass. Temporal and cross-candidate attention refine the expanded set, and reliability-constrained selection retains six joint predictions. On the Waymo Open Motion Dataset (WOMD) Interaction Prediction benchmark, JFER achieves a test Soft-mAP of 0.2868. Candidate-set analyses show that complementary exploration increases joint-future coverage and that part of this gain is retained under the original six-mode output budget, with larger gains at longer prediction horizons.

1 INTRODUCTION

Interactive motion forecasting predicts how multiple agents may move in a shared scene, providing downstream planning with possible future motions against which to evaluate driving actions (Ettinger et al., 2021; Shi et al., 2024). Motion prediction also supports integrated driving systems, including joint perception and planning (Hu et al., 2023), interactive planning (Huang et al., 2023), and unified simulation, prediction, and planning (Song et al., 2025). In interactive scenes, agents' future motions are coupled. At an intersection, for example, when a turning vehicle proceeds depends on whether an approaching vehicle slows down or continues through. The same observed history may therefore support several outcomes with different crossing orders and trajectories (Liu et al., 2024; Rowe et al., 2023). Representing these outcomes requires joint predictions that account for the behavior of both agents. To support real-time planning, however, the prediction set must remain compact, as evaluating more possible futures increases the cost of collision checking and assessing candidate driving actions. When several hypotheses describe similar intention combinations, fewer slots remain for other plausible interactions.

Existing methods construct joint predictions in different ways, as illustrated in Fig. 1. Agent-wise approaches combine marginal or conditional hypotheses by modeling cross-agent dependence (Sun et al., 2022; Luo et al., 2023; Rowe et al., 2023). Direct joint decoders generate scene-level futures from shared interaction representations (Seff et al., 2023; Wagner et al., 2024). Both approaches must determine which outcomes to represent within a limited prediction set. Selecting hypotheses primarily by confidence can favor several similar outcomes, while decoding every combination of agent-level intentions requires processing many candidates that will eventually be discarded. For instance, several pairs may differ slightly in endpoint position while describing the same crossing order, leaving an alternative yielding response absent from the output. This raises our first question: how can a small set of joint hypotheses be constructed to represent distinct, plausible combinations of agent intentions?

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.

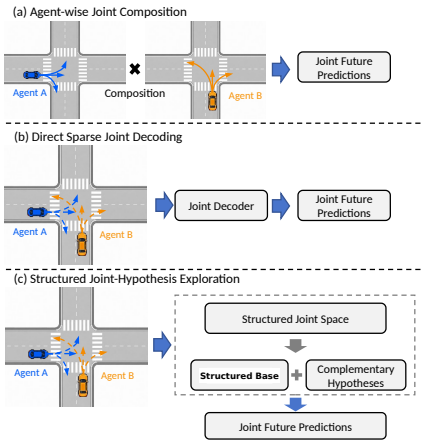


Figure 1: Representative formulations for compact joint prediction. JFER explores initial and complementary joint hypotheses.

An initial set may still omit plausible interactions. Adding hypotheses without checking its coverage can repeat existing outcomes, while selecting candidates solely for their differences can introduce alternatives with little support from the observed scene. Our second question is therefore: how can we identify plausible joint outcomes that the initial predictions miss and construct trajectories to represent them?

We introduce JFER, a Joint Future Exploration and Reasoning framework, to address these two questions. JFER first combines agent-level intentions into a joint-intention bank, where candidate pairs can be assessed before full trajectory decoding. Role-aware allocation assigns six prediction slots to distinct pairs while retaining a preference for each slot's initial intention association. This separates the number of intention combinations considered from the number of trajectories decoded.

Once the initial trajectories are available, complementary exploration selects additional intentions using their predicted support and distance from the decoded endpoints. To construct their temporal motions, JFER adapts existing joint hypotheses as motion donors through endpoint-directed deformation and bounded corrections. This reuses decoded motion structure without requiring a full decoding pass for each additional candidate.

We jointly refine the expanded candidate set by modeling the temporal evolution of each trajectory pair and exchanging information across candidates to update their geometry. A reliability-constrained consolidation strategy then selects six joint predictions from the refined pool, balancing prediction reliability and coverage of complementary interaction outcomes.

Our main contributions are:

- We propose structured joint-hypothesis construction that organizes agent-level intentions into an explicit bank and uses role-aware allocation to assign six prediction slots to distinct pairs, allowing a broader intention space to be considered without decoding every combination.
- We develop complementary exploration guided by predicted intention support and gaps in the decoded joint endpoints. Donor-based adaptation constructs additional trajectory pairs from existing predictions, followed by joint refinement and reliability-constrained selection under the original six-mode output budget.
- Experiments on WOMD Interaction Prediction demonstrate competitive performance. Ablations show that structured construction reduces prediction errors, while candidate-set analyses show that complementary exploration improves coverage of ground-truth joint futures in the final six predictions, with larger gains at longer horizons.

2 Related Work

2.1 Joint Motion Forecasting

Motion predictors encode roads and agent histories using vector representations and attention (Gao et al., 2020; Zhou et al., 2022; Nayakanti et al., 2023). Representation learning also benefits from masked pretraining (Cheng et al., 2023; Wagner et al., 2025) and training across datasets (Feng et al., 2024; Messaoud et al., 2025). Joint forecasting must also account for dependencies between agents' futures. Conditional approaches represent these dependencies during generation. M2I conditions a reactor's motion on an influencer's predicted trajectory, while FJMP factorizes decoding over a learned interaction graph (Sun et al., 2022; Rowe et al., 2023). Future-aware methods use anticipated intentions, lane occupancy, or motion to reason about interactions (Zhu et al., 2023; Park et al., 2023; Woo et al., 2024; Li et al., 2025). AgentFormer and Scene Transformer model temporal and cross-agent dependencies through attention (Yuan et al., 2021; Ngiam et al., 2022). Other formulations

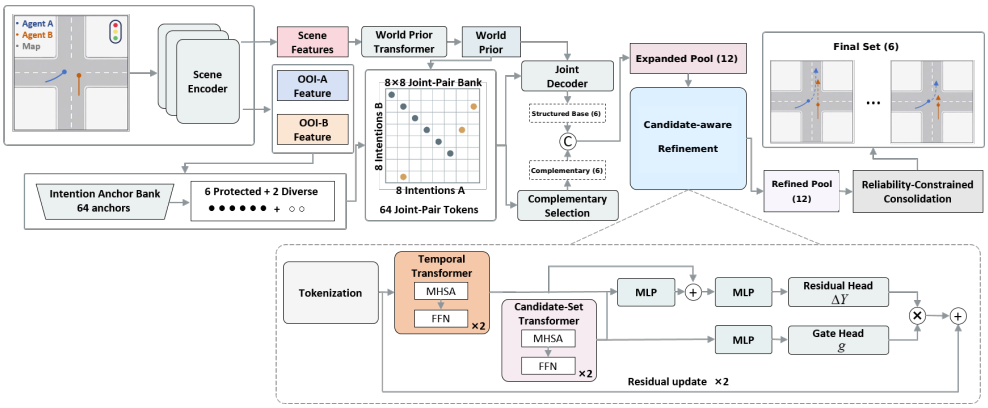


Figure 2: Overview of JFER. A structured joint-intention bank supports compact initial decoding and complementary expansion; the resulting candidate pool is jointly refined and consolidated back to the fixed output budget.

use autoregressive motion tokens (Seff et al., 2023; Wu et al., 2024), diffusion (Jiang et al., 2023), or agent-pair Gaussian covariance (Steiner et al., 2024). JFER allocates prediction slots over an explicit intention bank before decoding, separating the number of combinations considered from the number of trajectories generated.

2.2 Multimodal Hypothesis Modeling

Goals, anchors, and learned queries provide alternative futures. TNT and DenseTNT decode trajectories from selected goals (Zhao et al., 2021; Gu et al., 2021), while ProphNet combines proposals with anchor context (Wang et al., 2023). MTR localizes intentions and refines motion, and QCNet uses predicted proposals as adaptive anchors (Shi et al., 2022; Zhou et al., 2023). EDA evolves distinct anchors for training assignment, while DeMo separates intentions from dynamic states (Lin et al., 2024; Zhang et al., 2024). Annealed winner-takes-all training improves learning with few hypotheses (Xu et al., 2025). Parallel ModeSeq models dependencies among modes through causal masked attention (Zhou et al., 2026).

Predictions also guide subsequent computation. R-Pred retrieves context around proposals, and SmartRefine adapts refinement to prediction quality (Choi et al., 2023; Zhou et al., 2024). HPNet and RealMotion reuse previous predictions across observation windows (Tang et al., 2024; Song et al., 2024). MultiPath++ aggregates trajectory distributions into a compact output (Varadarajan et al., 2022), while Trajectory Mamba reduces computation through selective state-space modeling (Huang et al., 2025). JFER instead uses predicted support and endpoint coverage to select missing joint intentions, then adapts decoded pairs as motion donors.

3 Method

3.1 Problem Setup

We study joint forecasting for two interacting targets in an observed scene $\mathcal{S} = \{\mathcal{H}, \mathcal{M}, \mathcal{L}\}$, where $\mathcal{H}$, $\mathcal{M}$, and $\mathcal{L}$ denote motion histories, vectorized map elements, and observed traffic-control states. The model predicts K paired futures over T future steps,

$$\mathcal{P} = \left\{(\hat{\mathbf{Y}}_k, p_k)\right\}_{k=1}^{K}, \qquad \hat{\mathbf{Y}}_k = (\hat{\mathbf{Y}}_k^A, \hat{\mathbf{Y}}_k^B), \qquad \sum_{k=1}^{K} p_k = 1, \tag{1}$$

with $K = 6$ for WOMD interaction prediction. Each mode contains two trajectories and a joint probability. A joint intention specifies the two target endpoints, whereas a trajectory hypothesis describes both agents' motion over time. Similar endpoints may still correspond to different crossing times. Figure 2 summarizes the framework.

3.2 STRUCTURED CONSTRUCTION

Scene and world context. The MTR-based encoder extracts agent-history and map features. A world-context module learned during initialization supplies six observation-conditioned tokens from latent prior means. These tokens are fused with scene memory to form $\widetilde{\mathbf{E}}$ for joint-intention reasoning. The module remains frozen during subsequent candidate construction and refinement. Its future-conditioned posterior is used only during initialization (Appendix A.2).

Joint-intention bank. For each target, six scene- and world-conditioned prediction slots receive distinct assignments from 64 intention anchors. A role denotes a slot and its initial intention association. JFER retains these six anchors and adds two using marginal scores and endpoint diversity (Appendix A.3). Their Cartesian product gives

$$\mathcal{G} = \left\{ g_{ij} = (\mathbf{g}_i^A, \mathbf{g}_j^B) \mid i \in \mathcal{A}_A,\ j \in \mathcal{A}_B \right\}, \qquad |\mathcal{A}_A| = |\mathcal{A}_B| = 8, \quad |\mathcal{G}| = 64. \tag{2}$$

Each g_{ij} specifies two endpoints, not two decoded trajectories. Its pair token combines target-level intention features, relative endpoint geometry, and scene/world context,

$$\mathbf{b}_{ij} = f_{\text{pair}}\left(\mathbf{h}_i^A, \mathbf{h}_j^B, \boldsymbol{\phi}_{ij}, \mathbf{c}_{\mathcal{S}}, \widetilde{\mathbf{E}}\right), \tag{3}$$

where $\mathbf{h}_i^A$ and $\mathbf{h}_j^B$ are intention features, $\boldsymbol{\phi}_{ij}$ encodes relative endpoint geometry, and $\mathbf{c}_{\mathcal{S}}$ denotes scene context. A bank prior combines marginal intention evidence with learned pair compatibility.

Role-aware decoding. The two targets' initial anchors at slot k form its reference pair p_k^{ref}. Each slot attends to all 64 pair tokens. With slot feature $\mathbf{z}_k$ and attention dimension d, the assignment energy is

$$u_{kp} = \frac{\langle W_q \mathbf{z}_k, W_k \mathbf{b}_p \rangle}{\sqrt{d}} + \ell_p^{\text{bank}} + \beta\, \mathbb{I}[p = p_k^{\text{ref}}], \tag{4}$$

where ℓ_p^{bank} is the bank-prior logit. The bias favors the reference pair while allowing reassignment. Training uses straight-through hard/soft assignment, and inference greedily selects distinct pairs in slot order. Their tokens and endpoints adapt the initial queries through gated residual updates. The full joint decoder processes only these six selected queries, yielding the six paired trajectories of the structured base $\mathcal{C}_B$ and their scene-level joint scores. The other bank entries remain endpoint-level intentions rather than fully decoded hypotheses.

3.3 COMPLEMENTARY EXPANSION

Distances to the six decoded joint endpoints indicate bank intentions poorly covered by the structured base. JFER selects unused intention pairs that have predicted support but lie away from these endpoints, while avoiding repeated selections in the same region. For bank pair p, the selection score is

$$q(p) = z(\ell_p^{\text{bank}}) + \lambda_{\text{nov}} \psi(d_B(p)) + \lambda_{\text{div}} \psi(d_E(p)). \tag{5}$$

Here $z(\ell_p^{\text{bank}}) = (\ell_p^{\text{bank}} - \mu_{\mathcal{G}}) / \max(\sigma_{\mathcal{G}}, 0.25)$, where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are computed over all 64 bank logits in the current scene, and $\psi(d) = \tanh(d/d_e)$. Pairs already assigned to the structured base are masked only in the subsequent greedy complementary selection, not in the normalization. Coverage is assessed in the joint endpoint space of both targets. Let $\bar{\mathbf{g}}_p = [(\mathbf{g}_p^A)^\top, (\mathbf{g}_p^B)^\top]^\top$ and $\bar{\mathbf{y}}_{k,T} = [(\hat{\mathbf{Y}}_{k,T}^A)^\top, (\hat{\mathbf{Y}}_{k,T}^B)^\top]^\top$ denote the concatenated endpoints of candidate p and decoded hypothesis k. We define

$$d_B(p) = \min_{k \in \{1,\dots,6\}} \|\bar{\mathbf{g}}_p - \bar{\mathbf{y}}_{k,T}\|_2, \qquad d_E(p) = \min_{e \in \mathcal{E}_{\text{sel}}} \|\bar{\mathbf{g}}_p - \bar{\mathbf{g}}_e\|_2, \tag{6}$$

The base distance d_B measures the gap to the nearest decoded joint endpoint; d_E measures the gap to previously selected complementary pairs and is omitted for the first selection. The prior term retains intention evidence, while the two distances favor pairs missing from both the base and the growing complementary set. All three terms use observation-conditioned predictions; selection never accesses the ground-truth future.

A selected pair specifies a new joint endpoint but provides no temporal trajectory. Instead of decoding it from scratch, JFER retrieves the nearest base hypothesis in joint endpoint space as a motion

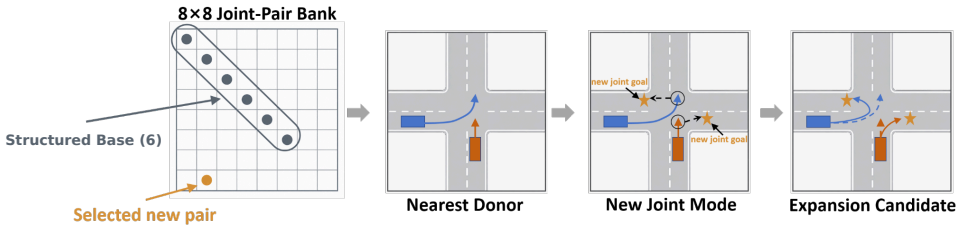


Figure 3: Coverage-directed complementary selection and donor-guided trajectory instantiation.

donor (Fig. 3). Let $\bar{\mathbf{Y}}_k(t) = [(\hat{\mathbf{Y}}_k^A(t))^\top, (\hat{\mathbf{Y}}_k^B(t))^\top]^\top$. For complementary intention e with joint endpoint $\bar{\mathbf{g}}_e$,

$$d(e) = \arg\min_{k\in\{1,\dots,6\}} \|\bar{\mathbf{g}}_e - \bar{\mathbf{y}}_{k,T}\|_2 , \tag{7}$$

Both targets share the donor selected by their concatenated endpoints in a common coordinate frame. The donor trajectory is adapted with a smooth endpoint-directed warp and bounded multi-horizon corrections,

$$\hat{\mathbf{Y}}_e^E(t) = \mathrm{sg}(\bar{\mathbf{Y}}_{d(e)}(t)) + \gamma_e h(t/T)\big(\bar{\mathbf{g}}_e - \bar{\mathbf{y}}_{d(e),T}\big) + \mathbf{r}_e(t), \tag{8}$$

where sg stops gradients, $h(\rho) = \rho^2(3-2\rho)$, and $\mathbf{r}_e(t)$ interpolates bounded corrections at 3, 5, and 8 s. The learned scalar gate $\gamma_e = \sigma(f_\gamma(\mathbf{h}_e)) \in (0,1)$ is computed from a summary of the three horizon tokens and shared by both agents and all future steps. Because the gate is not fixed to one and corrections remain bounded, the resulting endpoint need not coincide exactly with the selected bank endpoint. This adaptation reuses the donor's decoded temporal motion without another full decoding pass. Six selected pairs produce six complementary hypotheses $\mathcal{C}_E$, giving an expanded pool $\mathcal{C}^0 = \mathcal{C}_B \cup \mathcal{C}_E$ of twelve candidates.

3.4 Refinement and Consolidation

Candidate-aware refinement. For each of the twelve paired trajectories, sparse temporal tokens encode its motion across the prediction horizon. Within-candidate temporal attention updates these tokens; cross-candidate attention then exchanges features between alternative joint trajectories. The resulting features predict coordinate residuals, applied through a candidate-level gate in two bounded updates,

$$\mathbf{Y}_m^{r+1} = \mathbf{Y}_m^r + \frac{g_m^r}{2}\Delta_m^r, \qquad g_m^r = \sigma(\eta_m^r), \quad r \in \{0,1\}. \tag{9}$$

Trajectory features are recomputed after the first update, and the bounded residual branch is initialized near the identity. Refinement modifies trajectory coordinates only: it neither selects candidates nor updates confidence, selection, or admission scores.

Reliability-constrained consolidation. Consolidation retains the four base candidates with the highest selection logits and fills two remaining positions from the other base and complementary candidates. Complementary candidates must pass an admission threshold, exceed the replaceable-base score floor, and satisfy joint-endpoint redundancy suppression. Unfilled positions are backfilled from unselected base candidates. All candidates use their refined coordinates. Membership is determined before final confidence ordering (Appendix A.5).

With maximum replacement budget R,

$$\mathcal{I} = \mathrm{Guard}(\mathbf{Y}^2, \mathbf{s}, \mathbf{a}), \qquad |\mathcal{I}| = K, \qquad |\mathcal{I} \cap \mathcal{I}_E| \le R. \tag{10}$$

Here $\mathcal{I}_E$ indexes complementary candidates, $\mathbf{Y}^2$ denotes the refined pool after the second update, and the final set is $\mathcal{C}_F = \{\mathbf{Y}_i^2 \mid i \in \mathcal{I}\}$. The vector $\mathbf{s} \in \mathbb{R}^{12}$ contains selection logits for all candidates, whereas $\mathbf{a} \in \mathbb{R}^6$ contains admission logits only for the six complementary candidates; base candidates require no admission logits. We use $K = 6$ and $R = 2$, preserving at least four base identities and admitting at most two complementary candidates. Joint confidences are used to order the retained candidates and are normalized to form the probabilities of the final set. Admission decisions therefore remain separate from the reported output probabilities.

3.5 OPTIMIZATION

JFER follows a two-stage, module-wise protocol. After initialization, Stage I learns candidate construction, scoring, and complementary expansion, followed by confidence calibration. Stage II freezes these modules and optimizes geometry refinement alone. Appendix A.6 details the supervision.

Candidate construction is supervised by target-level endpoint assignments and joint-pair quality. Coverage and uniqueness objectives encourage the six slots to represent distinct intention pairs. Complementary candidates receive trajectory supervision and admission targets based on whether they improve on replaceable base predictions or match futures missed by the base at a supervised horizon. Confidence calibration then fixes trajectory geometry and optimizes candidate ranking. The corresponding objectives are detailed in Appendix A.6.

At 3, 5, and 8 seconds, refinement supervision is assigned according to pre-refinement candidate quality. The geometry objective is summarized as

$$\mathcal{L}_{\text{geom}} = \mathcal{L}_{\text{fit}} + \mathcal{L}_{\text{pres}} + \mathcal{L}_{\text{reg}}, \tag{11}$$

where $\mathcal{L}_{\text{fit}}$ supervises bounded correction and pool coverage, $\mathcal{L}_{\text{pres}}$ discourages increases in the set's best matching cost and loss of candidate diversity, and $\mathcal{L}_{\text{reg}}$ regularizes residual magnitude and temporal smoothness. Ground-truth futures are used only for training supervision.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. We evaluate JFER on the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021) under the official Interaction Prediction protocol. WOMD contains approximately 487K training, 44K validation, and 44K test scenarios. We use approximately 204K interaction scenes from the official training split. Given 1 s of observed agent motion, map context, and observed traffic controls, we predict six joint futures for two interacting agents over the next 8 s.

Evaluation metrics. We use the official WOMD evaluator and report Soft-mAP, mAP, minimum average displacement error (minADE), minimum final displacement error (minFDE), miss rate (MR), and overlap rate (Overlap). Soft-mAP is the primary ranking metric. Results are averaged over the 3, 5, and 8 s prediction horizons and the vehicle, pedestrian, and cyclist categories. We also measure joint-future coverage, endpoint duplication, and oracle candidate-set quality to analyze the effects of candidate construction, expansion, and selection. These diagnostic metrics are defined in Appendix B.

Implementation details. We adopt MTR (Shi et al., 2022) as the scene encoder and implement JFER in PyTorch. Training uses AdamW on five NVIDIA RTX PRO 6000 GPUs with a global batch size of 80. We evaluate on the official validation and test splits. All ablation settings retain six joint predictions for official evaluation. The training protocol and inference settings are provided in Appendix A.

4.2 MAIN RESULTS

Overall performance. Table 1 compares JFER with representative methods on WOMD Interaction Prediction (Shi et al., 2024; Liu et al., 2024; Sun et al., 2026; Zhou et al., 2026; Yan et al., 2025; Wagner et al., 2026). On the official test set, JFER achieves a Soft-mAP of 0.2868 and ranks second among the listed methods in Soft-mAP, mAP, minADE, and minFDE. Its performance is competitive across both confidence-sensitive metrics and displacement errors, although its MR remains higher than those of MTR++ and Parallel ModeSeq. On the validation set, JFER achieves the highest reported mAP and the lowest minADE and minFDE among the compared methods.

Category-wise results. Table 2 reports results by agent category. Vehicles achieve the highest Soft-mAP and mAP. Pedestrian predictions have the lowest displacement errors but the highest overlap rate, indicating that positional accuracy alone does not ensure fewer overlaps. Cyclists have the highest minFDE and MR, reflecting greater difficulty in matching their future endpoints.

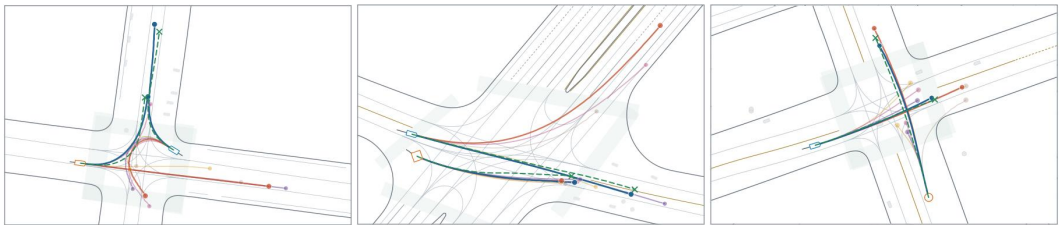
Table 1: Comparison on WOMD Interaction Prediction. Test baselines are official leaderboard submissions; validation results are taken from the original papers. "–" denotes an unreported metric. Best and second-best results per split are shown in **bold** and underline.

Split	Method	minADE ↓	minFDE ↓	MR ↓	mAP ↑	Soft-mAP ↑
Test	MTR++	0.8795	1.9505	0.4143	0.2326	0.2368
	BeTopNet	0.9744	2.2744	0.4355	0.2412	0.2466
	IMPACT	0.9738	2.2734	0.4316	0.2659	0.2718
	Parallel ModeSeq	**0.7707**	**1.6897**	**0.3782**	**0.2949**	**0.2978**
	JFER (Ours)	0.8692	1.9416	0.4248	0.2804	0.2868
Val	MTR++	0.8859	1.9712	**0.4106**	0.2398	–
	BeTopNet	0.9304	2.1340	0.4154	0.2366	–
	TrajFlow	0.9273	2.1249	0.4110	0.2464	0.2531
	RetroMotion	0.8718	1.9383	0.4216	0.2515	–
	JFER (Ours)	**0.8641**	**1.9152**	0.4144	**0.2799**	**0.2868**

Table 2: Category-wise JFER performance on the official WOMD Interaction Prediction test set, averaged over prediction horizons.

Object type	Soft-mAP ↑	mAP ↑	minADE ↓	minFDE ↓	MR ↓	Overlap ↓
Vehicle	0.3906	0.3839	0.9319	2.0733	0.3525	0.1118
Pedestrian	0.2946	0.2861	0.6693	1.4853	0.3986	0.2754
Cyclist	0.1753	0.1713	1.0065	2.2662	0.5233	0.1260
Overall	0.2868	0.2804	0.8692	1.9416	0.4248	0.1711

Qualitative results. Figure 4 presents the final six joint predictions in three representative scenes. In (a), the trajectories branch toward different outgoing directions at the intersection. In (b), several trajectories initially follow similar paths and separate farther along the road, producing alternatives with different curvature and endpoint locations. In (c), the predictions include motion along the road and crossing movements near the junction. Each mode specifies a trajectory for both agents, so these variations represent different paired outcomes. The examples show how the six predictions accommodate distinct route choices as well as variation within a shared route.



(a) (b) (c)

Figure 4: The six-mode final set in three interaction scenes: (a) vehicle–vehicle, (b) vehicle–cyclist, and (c) vehicle–pedestrian. Each color denotes one joint trajectory pair, and lower-probability pairs are shown with greater transparency.

4.3 Ablation Studies

Effect of structured construction. Table 3 compares direct joint decoding with structured construction under the same six-mode budget. Selecting intention pairs before decoding reduces minADE and minFDE and lowers MR, indicating that the six predictions both fit future trajectories more closely and miss fewer joint outcomes. These gains are obtained without increasing the number of decoded trajectories. The accompanying improvement in confidence-sensitive metrics suggests that the selected pairs also support better ranking of the resulting trajectories. This comparison tests the construction strategy as a whole, so it does not isolate the source of each gain. The next ablation examines the allocation rule and the size of the intention bank separately.

Table 3: Structured construction.

Metric	Direct	Structured Base
Soft-mAP ↑	0.1996	0.2727
mAP ↑	0.1918	0.2667
MR ↓	0.4732	0.4165
minADE ↓	0.9991	0.8637
minFDE ↓	2.2820	1.9115

Table 4: Role-aware allocation (RA) and bank size.

Metric	w/o RA M=8	w/ RA M=6	w/ RA M=7	w/ RA M=8
Soft-mAP ↑	0.2698	0.2712	0.2717	0.2727
mAP ↑	0.2638	0.2651	0.2656	0.2667
MR ↓	0.4157	0.4167	0.4140	0.4165
minADE ↓	0.8667	0.8667	0.8658	0.8637
minFDE ↓	1.9209	1.9209	1.9180	1.9115

Table 5: Module ablations with expansion and six outputs. The last row is JFER.

Refinement	Consolidation	Soft-mAP ↑	mAP ↑	MR ↓	minADE ↓	minFDE ↓
—	—	0.2818	0.2749	0.4181	0.8679	1.9222
✓	—	0.2844	0.2762	0.4167	0.8668	1.9185
✓	✓	0.2868	0.2799	0.4144	0.8641	1.9152

Table 6: Candidate-set coverage, endpoint duplication, and Oracle Soft-mAP across JFER stages.

Hypothesis set	#Modes	Oracle Soft-mAP ↑	Cov.@3s ↑	Cov.@5s ↑	Cov.@8s ↑	Dup.@8s ↓
Structured Base	6	0.4012	0.7294	0.6443	0.5254	0.0771
Expanded Pool	12	0.4286	0.7517	0.6717	0.5578	0.2017
Refined Pool	12	0.4291	0.7524	0.6729	0.5592	0.1898
Final Set	6	0.4054	0.7326	0.6490	0.5333	0.0859

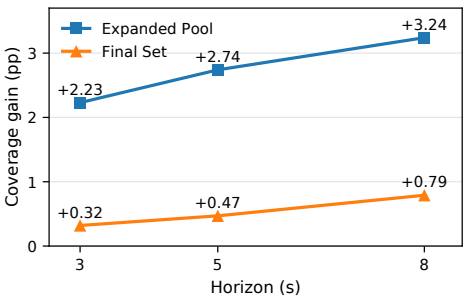


Figure 5: Coverage gain over the structured base across prediction horizons.

Effect of allocation and intention-bank size. Table 4 varies the number of intentions per agent (M) and compares role-aware allocation with selection without RA, which retains distinct pairs but removes slot-pair compatibility and reference bias. At eight intentions per agent, role-aware allocation improves Soft-mAP, mAP, and displacement errors, with a slightly higher MR. With six decoded pairs throughout, enlarging the bank from six to eight intentions improves these same four metrics. This supports considering more intention combinations before decoding. MR is lowest at seven, so the eight-intention setting does not dominate every metric.

Effect of individual components. Table 5 evaluates refinement and consolidation with expansion retained. Expansion alone improves Soft-mAP over the structured base but increases displacement errors and MR. A larger pool alone therefore does not ensure more accurate selected trajectories. Refinement improves all metrics with scores fixed, so the improvement comes from geometry updates. Consolidation improves all five metrics over selecting the six highest-scored candidates from the same refined pool. Compared with structured base, JFER improves Soft-mAP, mAP, and MR, with slightly higher minADE and minFDE.

Effect on coverage and redundancy. Table 6 tracks the candidate sets across stages. Expansion increases coverage and Oracle Soft-mAP, but also endpoint duplication. Oracle results involve different candidate counts and are not directly comparable deployment scores. Refinement increases coverage and reduces duplication. The final set retains higher coverage than structured base, with slightly higher duplication. The added matches therefore cannot be explained simply by lower endpoint redundancy. Appendix B.2 confirms that consolidation recovers more matches than it loses relative to retaining the six refined base candidates.

Effect across prediction horizons. Figure 5 shows larger coverage gains at longer horizons. The final six predictions retain part of the gain at every horizon, with the largest improvement at 8 s. The gap between pool and final coverage shows that some recovered outcomes are lost during consolidation.

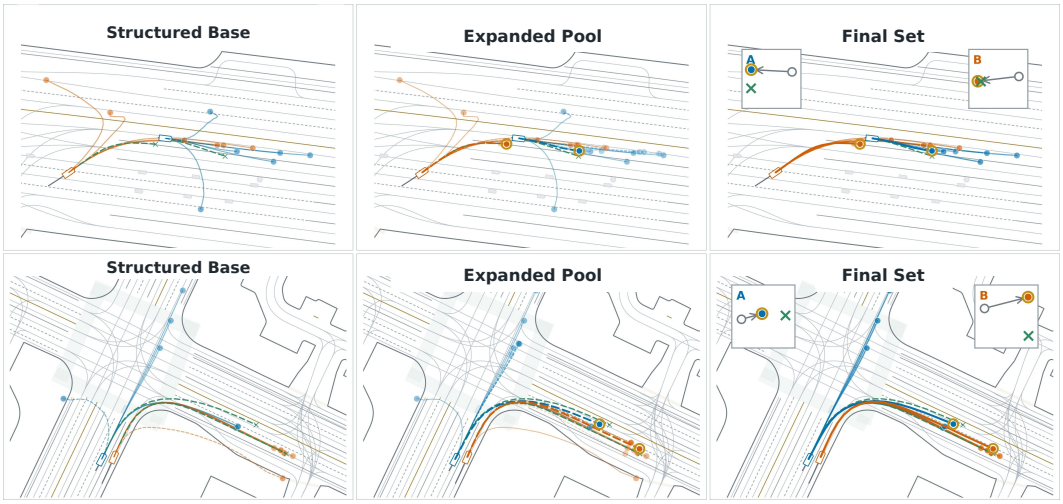


Figure 6: Trajectory exploration and consolidation in JFER. Top: recovery of an under-represented branch; bottom: better-aligned complementary replacement.

Qualitative effects of exploration and selection. Figure 6 illustrates how complementary candidates change the prediction set. In the first example, expansion adds a trajectory pair to a turning branch insufficiently represented by the base, and final selection retains this alternative. In the second, the maneuver is already represented, but expansion provides a trajectory pair better aligned with the ground truth. These examples illustrate two sources of improvement: recovering an omitted branch and improving the trajectory geometry within an existing branch.

5 Conclusion

We presented JFER for interactive motion forecasting under a fixed prediction budget. JFER explicitly organizes joint intentions to construct initial predictions and uses their coverage in joint endpoint space to guide complementary exploration. Experiments on WOMD Interaction Prediction demonstrate competitive performance. Candidate-set analyses show that complementary candidates recover joint futures missed by the initial predictions, with coverage gains retained in the final six outputs and larger benefits at longer prediction horizons. Endpoint-based intention selection may overlook interaction outcomes that reach similar destinations but differ in crossing order or timing. Future work will incorporate explicit interaction timing into complementary intention selection.

AI use statement

Generative AI tools were used to assist with language polishing and LaTeX typesetting and layout. All scientific ideas, method design, experiments, analyses, and conclusions were developed and verified by the authors.

Reproducibility statement

Appendix A describes the model components, training protocol, and inference procedure. Appendix B defines the diagnostic metrics and reports additional controlled experiments, and Appendix C provides qualitative examples. Evaluation uses the official WOMD validation and test splits and evaluator. The anonymized implementation is available at the link below.

Anonymous code: `https://anonymous.4open.science/r/JFER`

REFERENCES

Jie Cheng, Xiaodong Mei, and Ming Liu. Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders. In *IEEE/CVF International Conference on Computer Vision*, pp. 8679–8689, 2023.

Sehwan Choi, Jungho Kim, Junyong Yun, and Jun Won Choi. R-Pred: Two-stage motion prediction via tube-query attention-based trajectory refinement. In *IEEE/CVF International Conference on Computer Vision*, pp. 8525–8535, 2023.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The Waymo Open Motion Dataset. In *IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud, Eloi Zablocki, Matthieu Cord, and Alexandre Alahi. UniTraj: A unified framework for scalable vehicle trajectory prediction. In *European Conference on Computer Vision*, pp. 106–123, 2024.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. VectorNet: Encoding HD maps and agent dynamics from vectorized representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525–11533, 2020.

Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *IEEE/CVF International Conference on Computer Vision*, pp. 15303–15312, 2021.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023.

Yizhou Huang, Yihua Cheng, and Kezhi Wang. Trajectory Mamba: Efficient attention-Mamba forecasting model based on selective SSM. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12058–12067, 2025.

Zhiyu Huang, Haochen Liu, and Chen Lv. GameFormer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *IEEE/CVF International Conference on Computer Vision*, pp. 3903–3913, 2023.

Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, and Dragomir Anguelov. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2023.

Shijie Li, Chunyu Liu, Xun Xu, Si Yong Yeo, and Xulei Yang. Future-aware interaction network for motion forecasting. In *IEEE/CVF International Conference on Computer Vision*, pp. 7505–7515, 2025.

Longzhong Lin, Xuewu Lin, Tianwei Lin, Lichao Huang, Rong Xiong, and Yue Wang. EDA: Evolving and distinct anchors for multimodal motion prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3432–3440, 2024.

Haochen Liu, Li Chen, Yu Qiao, Chen Lv, and Hongyang Li. Reasoning multi-agent behavioral topology for interactive autonomous driving. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92605–92637, 2024.

Wenjie Luo, Cheol Park, Andre Cornman, Benjamin Sapp, and Dragomir Anguelov. JFP: Joint future prediction with interactive multi-agent modeling for autonomous driving. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1457–1467, 2023.

Kaouther Messaoud, Matthieu Cord, and Alexandre Alahi. Towards generalizable trajectory prediction using dual-level representation learning and adaptive prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27564–27574, 2025.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple and efficient attention networks. In *IEEE International Conference on Robotics and Automation*, pp. 2980–2987, 2023.

Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Ben Sapp, Zhifeng Chen, and Jonathon Shlens. Scene Transformer: A unified architecture for predicting multiple agent trajectories. In *International Conference on Learning Representations*, 2022.

Daehee Park, Hobin Ryu, Yunseo Yang, Jegyeong Cho, Jiwon Kim, and Kuk-Jin Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *International Conference on Learning Representations*, 2023.

Luke Rowe, Martin Ethier, Eli-Henry Dykhne, and Krzysztof Czarnecki. FJMP: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13745–13755, 2023.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *IEEE/CVF International Conference on Computer Vision*, pp. 8579–8590, 2023.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6531–6543, 2022.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. MTR++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3955–3971, 2024.

Nan Song, Bozhou Zhang, Xiatian Zhu, and Li Zhang. Motion forecasting in continuous driving. In *Advances in Neural Information Processing Systems*, volume 37, pp. 78147–78168, 2024.

Nan Song, Junzhe Jiang, Jingyu Li, Xiatian Zhu, and Li Zhang. UniMotion: A unified motion framework for simulation, prediction and planning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 135052–135074, 2025.

Marlon Steiner, Marvin Klemp, and Christoph Stiller. MAP-Former: Multi-agent-pair gaussian joint prediction. In *IEEE Intelligent Vehicles Symposium*, pp. 758–764, 2024.

Jiawei Sun, Xibin Yue, Jiahui Li, Tianle Shen, Chengran Yuan, Shuo Sun, Sheng Guo, Quanyun Zhou, and Marcelo H. Ang, Jr. IMPACT: Behavioral intention-aware multimodal trajectory prediction with adaptive context trimming. *IEEE Robotics and Automation Letters*, 11(1):610–617, 2026.

Qiao Sun, Xin Huang, Junru Gu, Brian C. Williams, and Hang Zhao. M2I: From factored marginal trajectory prediction to interactive prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6543–6552, 2022.

Xiaolong Tang, Meina Kan, Shiguang Shan, Zhilong Ji, Jinfeng Bai, and Xilin Chen. HPNet: Dynamic trajectory forecasting with historical prediction attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15261–15270, 2024.

Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S. Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, and Benjamin Sapp. MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *IEEE International Conference on Robotics and Automation*, pp. 7814–7821, 2022.

Royden Wagner, Omer Sahin Tas, Marlon Steiner, Fabian Konstantinidis, Hendrik Konigshof, Marvin Klemp, Carlos Fernandez, and Christoph Stiller. SceneMotion: From agent-centric embeddings to scene-wide forecasts. In *IEEE International Conference on Intelligent Transportation Systems*, pp. 812–818, 2024.

Royden Wagner, Omer Sahin Tas, Marvin Klemp, and Carlos Fernandez. JointMotion: Joint self-supervision for joint motion prediction. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3395–3406, 2025.

Royden Wagner, Omer Sahin Tas, Felix Hauser, Marlon Steiner, Dominik Strutz, Abhishek Vivekanandan, Jaime Villa, Yinzhe Shen, Carlos Fernandez, and Christoph Stiller. RetroMotion: Retrocausal motion forecasting models are instructable. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4302–4311, 2026.

Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. ProphNet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21995–22003, 2023.

Sungmin Woo, Minjung Kim, Donghyeong Kim, Sungjun Jang, and Sangyoun Lee. FIMP: Future interaction modeling for multi-agent motion prediction. In *IEEE International Conference on Robotics and Automation*, pp. 14457–14463, 2024.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 114048–114071, 2024.

Yihong Xu, Victor Letzelter, Mickaël Chen, Éloi Zablocki, and Matthieu Cord. Annealed winner-takes-all for motion forecasting. In *IEEE International Conference on Robotics and Automation*, pp. 1264–1270, 2025.

Qi Yan, Brian Zhang, Yutong Zhang, Daniel Yang, Joshua White, Di Chen, Jiachao Liu, Langechuan Liu, Binnan Zhuang, Shaoshuai Shi, and Renjie Liao. TrajFlow: Multi-modal motion prediction via flow matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3923–3928, 2025.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M. Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *IEEE/CVF International Conference on Computer Vision*, pp. 9813–9823, 2021.

Bozhou Zhang, Nan Song, and Li Zhang. DeMo: Decoupling motion forecasting into directional intentions and dynamic states. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106582–106606, 2024.

Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. TNT: Target-driven trajectory prediction. In *Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 895–904, 2021.

Yang Zhou, Hao Shao, Letian Wang, Steven L. Waslander, Hongsheng Li, and Yu Liu. SmartRefine: A scenario-adaptive refinement framework for efficient motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15281–15290, 2024.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. HiVT: Hierarchical vector transformer for multi-agent motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023.

Zikang Zhou, Haibo Hu, Xinhong Chen, Yifan Zhang, Nan Guan, Yung-Hui Li, Chun Jason Xue, and Jianping Wang. Mode-as-Sequence: Translating multimodal motion prediction into unified sequential mode modeling. *arXiv preprint arXiv:2605.24037*, 2026.

Yiyao Zhu, Di Luan, and Shaojie Shen. BiFF: Bi-level future fusion with polyline-based coordinate for interactive trajectory prediction. In *IEEE/CVF International Conference on Computer Vision*, pp. 8260–8271, 2023.

A Additional Implementation Details

This appendix specifies the construction and supervision of JFER's candidate sets, including the world context, intention assignments, and final selection rule.

A.1 Inference Pipeline

JFER constructs the six-mode structured base $\mathcal{C}_B$ and adds six complementary hypotheses $\mathcal{C}_E$ to form the expanded pool. Refinement produces the twelve-candidate refined pool $\mathcal{C}_R$, and consolidation returns the six-mode final set $\mathcal{C}_F$. The 8×8 joint-intention bank contains endpoint pairs. Only the six pairs assigned to prediction slots undergo full trajectory decoding. Complementary trajectories reuse decoded motion through donor adaptation.

A.2 World-Context Representation

Initialization and inference. The world-context module is learned within JFER's staged training pipeline. For each of the two targets, center, object, map, and current-state features provide four scene tokens. A scene Transformer encodes these tokens, whose mean forms the scene representation $\mathbf{s}$. Six learned embeddings $\mathbf{e}_w$ are conditioned on $\mathbf{s}$ and processed jointly by a world-slot Transformer. A prior head predicts a diagonal Gaussian for each slot. The tokens used downstream are deterministic projections of the prior means,

$$\mathbf{u}_w = \mathrm{WorldEnc}(\mathbf{e}_w + \mathbf{s}), \qquad \mathbf{z}_w^p = \mathbf{u}_w + f_z(\boldsymbol{\mu}_w^p), \qquad w = 1, \ldots, 6. \tag{12}$$

These tokens are fused with scene memory before joint-intention reasoning. No latent sampling is performed. The world-context module is frozen during subsequent candidate-bank training and refinement.

Training-only posterior. Sparsely sampled ground-truth futures provide position, velocity, and validity features. A future Transformer and masked mean pooling produce a future representation, which is combined with each $\mathbf{u}_w$ to predict posterior Gaussian parameters. Posterior tokens use the same projection f_z applied to posterior means. Shared rollout dynamics map prior and posterior tokens to sparse positions and velocities for both targets.

Posterior rollouts are evaluated using a weighted combination of position and velocity errors. A softmax over their negative errors assigns greater responsibility to slots that better match the observed future. Initialization uses prior rollout supervision, diagonal-Gaussian KL regularization in the direction $q\|p$, and soft cross-entropy for world probabilities. Auxiliary terms encourage endpoint diversity, balanced slot use, and consistency between prediction modes and world slots. Posterior rollouts supply responsibility targets without a separate rollout-fitting objective. The posterior is absent at inference.

A.3 Structured Base Construction

Initial anchor associations. Each target has 64 type-specific intention anchors. Six scene- and world-conditioned role slots predict target-specific anchor assignment logits. Here a role denotes a prediction slot and its intention association. Slot-ordered greedy assignment selects a different anchor for each slot of the same target. These six anchors provide the initial intentions retained in the bank.

Additional intentions. Let m_i be the scene-conditioned marginal logit of anchor $\mathbf{g}_i$, and let $\mathcal{A}$ initially contain the six assigned anchors for one target. Two further anchors are added successively by

$$i^* = \arg\max_{i \notin \mathcal{A}} \left[m_i + \lambda_m \tanh\left(\frac{\min_{j \in \mathcal{A}} \|\mathbf{g}_i - \mathbf{g}_j\|_2}{d_m} \right) \right], \tag{13}$$

where λ_m controls the contribution of anchor separation and d_m sets its distance scale. The selected index is added to $\mathcal{A}$ before the next step. Marginal logits are detached for this discrete selection. This operation enlarges the intention bank before base decoding, independently of the later complementary trajectory expansion.

Reference pairs and assignment. For each target, the first six retained anchors are ordered by their initial slot assignments. Pairing the two targets' anchors at the same slot yields six reference pairs. The Cartesian product of the eight retained intentions per target forms 64 pairs. Each slot attends to the entire bank, with a bias toward its reference pair. Training uses straight-through assignment, with hard choices in the forward pass and soft assignments for gradients. Inference selects distinct pairs greedily in slot order. Gated residual adaptation incorporates the selected tokens and endpoints into the queries before the six joint trajectories are decoded.

A.4 Complementary Expansion

The selection score in Eq. (5) combines bank support, distance from decoded base endpoints, and distance from previously selected complementary intentions. Bank logits are normalized across all entries before base-assigned pairs are masked for selection. The within-expansion distance is omitted for the first selection.

Each selected intention retrieves the base pair with the nearest concatenated joint endpoint in the shared scene frame. Both targets reuse motion from the same donor. The scalar warp gate is shared across agents and future steps. Residual corrections are interpolated from knots at the three evaluation horizons, with bounds that increase with the prediction horizon. Together with the learned warp gate, these corrections adapt the donor without enforcing exact arrival at the selected endpoint. Six such pairs augment the six base hypotheses.

A.5 Candidate Refinement and Consolidation

Geometry refinement. Each paired trajectory is represented by sparse temporal tokens. Temporal attention encodes its evolution, and attention across candidates supplies information about the other hypotheses. These features predict gated coordinate residuals within horizon-dependent bounds. Two successive updates recompute trajectory features between steps. Candidate trajectories, world and spatial evidence, and scores are detached at the refinement input. Confidence, selection, and admission scores remain unchanged.

Final candidate selection. Selection logits determine candidate membership, admission logits provide an additional test for complementary candidates, and joint-confidence logits determine the final probabilities. Consolidation first retains the four refined base candidates with the highest selection logits. It then scans the two remaining base candidates and six complementary candidates in descending selection-logit order. A complementary candidate must have admission probability at least 0.60, exceed the replaceable-base selection-score floor by 0.05, and pass endpoint redundancy suppression. At most two complementary candidates are admitted.

Suppression uses Euclidean distance between concatenated 8 s endpoints of both targets, with a threshold of 2.0 m. Unfilled positions are backfilled from unselected base candidates in base ranking order, without another suppression test. The six retained candidates are reordered by joint-confidence logits, and a softmax over these six logits gives their output probabilities.

A.6 Training Protocol

After scene and world-context initialization, Stage I trains candidate construction, scoring, complementary expansion, and confidence calibration in successive phases of 3, 6, 6, and 3 epochs. The selected checkpoint from each phase initializes the next. The scene encoder, base decoder, and world-context module remain frozen. Stage II trains geometry refinement for 6 epochs with proposal generation and all scoring heads fixed. We use AdamW with an initial learning rate of 10^{-4}.

Candidate construction. For each target, the marginal head is supervised by cross-entropy against the anchor nearest to the final valid ground-truth endpoint. Pair quality is the mean endpoint error of the two targets. A softmax over negative pair errors supplies soft cross-entropy targets for the pair head. A coverage term aligns the aggregate slot-assignment distribution with this pair target, and a uniqueness term penalizes similar assignment distributions across slots.

Expansion and confidence. Expansion training combines trajectory regression, coverage and diversity objectives with admission, horizon-matching, ranking, and Soft-mAP credit supervision. An admission target is positive if the complementary candidate improves the weighted ground-truth cost over the replaceable-base boundary by at least 0.05, or matches a valid horizon missed by all base candidates. Horizon targets record joint matching at 3, 5, and 8 s. Within-expansion ranking uses soft targets derived from negative weighted costs. Score regularization limits changes in the predicted scores.

Confidence calibration freezes trajectory geometry and optimizes Soft-mAP credit, cross-ranking between base and complementary candidates, and score regularization. Admission and horizon-matching supervision are inactive in this phase. Consequently, admission scores are inherited from expansion training, while final joint confidences are calibrated separately.

Refinement supervision. At each supervised horizon, the pre-refinement candidate with the lowest ground-truth cost receives responsibility. The fitting, preservation, and regularization groups in Eq. (11) contain

$$\mathcal{L}_{\text{fit}} = \lambda_f \mathcal{L}_{\text{flow}} + \lambda_g \mathcal{L}_{\text{gate}} + \lambda_a \mathcal{L}_{\text{winnerADE}} + \lambda_c \mathcal{L}_{\text{coverage}}. \tag{14}$$

$$\mathcal{L}_{\text{pres}} = \lambda_n \mathcal{L}_{\text{no-harm}} + \lambda_d \mathcal{L}_{\text{diversity}}. \tag{15}$$

$$\mathcal{L}_{\text{reg}} = \lambda_r \mathcal{L}_{\text{residual}} + \lambda_s \mathcal{L}_{\text{smooth}}. \tag{16}$$

The coefficients λ are fixed during refinement. The flow target is a bounded displacement from the original candidate toward the ground truth. Gate, winner-ADE, and coverage supervision guide its application. The no-harm term penalizes increases in the candidate set's best cost at each horizon, and the diversity term discourages contraction of pre-refinement inter-candidate distances. Residual regularization limits correction magnitude, while temporal regularization penalizes second-order changes. These objectives update geometry without changing candidate scores.

B Additional Experimental Details

This appendix defines the candidate-set diagnostics and provides additional controlled analyses of temporary expansion, candidate replacement, pair selection, and module overhead. Official WOMD metrics follow the benchmark evaluator and are not redefined here.

B.1 Evaluation Protocol and Metrics

JFER decodes six structured base hypotheses, expands them to twelve candidates, and refines this pool before consolidation returns six outputs. Official WOMD metrics use six predictions. Expansion and refinement ablations retain the six highest-scored candidates from their respective pools, while JFER applies the consolidation rule in Appendix A.5. Pool-level diagnostics evaluate each named set directly.

For the structured-construction comparison in Table 3, Direct uses the same scene encoder, joint decoder, training protocol, and six-output budget as the Structured Base, but bypasses the joint-intention bank and role-aware pair assignment and directly decodes the six prediction queries.

Table 4 evaluates the six base predictions under alternative allocation rules. Retaining six, seven, or eight intentions per agent produces banks of 36, 49, or 64 pairs, respectively. These comparisons precede complementary expansion.

For a candidate set $\mathcal{C}$ and prediction horizon t, we use retrospective diagnostics to characterize support, redundancy, and confidence ordering. These quantities are used only for analysis and do not alter inference.

Joint future coverage. Coverage at horizon t, denoted $\mathrm{Cov.@}t$, is the fraction of scenes for which at least one joint hypothesis satisfies the same class-dependent matching criteria used by the WOMD evaluator. It measures whether a matching future is present in the candidate set, independently of how candidate confidence is ordered.

Endpoint duplication. For M candidates, endpoint duplication is the fraction of unordered candidate pairs whose mean endpoint distance across the two target agents falls below a fixed threshold,

$$D_t^{\text{end}}(i,j) = \frac{1}{2} \sum_{a \in \{A,B\}} \left\| \hat{\mathbf{Y}}_{i,a,t} - \hat{\mathbf{Y}}_{j,a,t} \right\|_2, \qquad \text{Dup.@}t = \frac{\sum_{i<j} \mathbb{I}[D_t^{\text{end}}(i,j) < \delta_{\text{dup}}]}{\binom{M}{2}}. \quad (17)$$

We use $\delta_{\text{dup}} = 2.0$ m throughout the diagnostic analysis. Lower values indicate less endpoint redundancy within the candidate set. Coverage is evaluated separately and is not inferred from duplication. This metric is also distinct from the joint-endpoint suppression used during final consolidation.

Oracle candidate-set quality. Oracle Soft-mAP keeps candidate trajectories and the set of confidence magnitudes fixed, while permuting the association between them using ground-truth matching quality. Candidates are ranked by their weighted joint-match score across valid horizons, with lower weighted displacement cost breaking ties and candidate index resolving any remaining ties. The original confidence values, sorted in descending order, are assigned to candidates in this order before Soft-mAP is recomputed.

This diagnostic evaluates the effect of ground-truth-guided confidence ordering on a fixed trajectory set. Pool-12 values use all twelve internal candidates, without selecting six first. Final-6 values use only the six candidates returned by consolidation. Comparisons across these counts characterize internal candidate sets and are distinct from fixed-budget official evaluation.

Rescue and harm. For the paired replacement analysis, both views start from the same refined pool. The base reference retains the six refined base candidates, while the alternative applies the final consolidation rule. A rescue occurs when consolidation introduces a matching 8 s future absent from the refined base set; a harm is the reverse. This isolates the effect of candidate membership from trajectory refinement.

B.2 Additional Ablation Studies

Effect of the exploration budget. Table 7 varies the temporary complementary budget at a fixed checkpoint and inference setting. With zero complementary candidates, downstream processing remains active, so this control differs from the structured base. Official metrics and coverage use the final six predictions, and duplication is computed on the refined pool.

Table 7: Temporary complementary budget E. Coverage and official metrics use the final six predictions. Duplication uses the refined pool.

E	Pool	Soft-mAP ↑	minFDE ↓	Cov.@8s ↑	Dup.@8s ↓
0	6	0.2819	1.9250	0.5259	0.0773
2	8	0.2847	1.9202	0.5298	0.1320
4	10	0.2858	1.9170	0.5321	0.1658
6	12	0.2868	1.9152	0.5333	0.1898

Over the tested budgets, additional complementary candidates improve final Soft-mAP, minFDE, and 8 s coverage. Duplication also rises within the refined pool. Thus, a larger internal pool can improve the selected six predictions even when some additional candidates have similar endpoints.

Effect of complementary intention selection. Table 8 isolates the complementary intention-selection rule under the same checkpoint and temporary candidate budget. The prior-only variant uses normalized bank logits. The second adds distance from base endpoints, and the full rule also uses distance from previously selected complementary intentions. The temporary budget is $E = 6$ for all three variants.

Table 8: Ablation of complementary pair selection. Dup.@8s is measured on the refined pool.

Selection	Soft-mAP ↑	minFDE ↓	Cov.@8s ↑	Dup.@8s ↓
Prior only	0.2833	1.9200	0.5290	0.2330
Prior + novelty	0.2841	1.9171	0.5317	0.2084
Full selection	0.2868	1.9152	0.5333	0.1898

Adding base-relative novelty improves final coverage, and the full selection rule further improves coverage while reducing duplication in the refined pool. The comparison supports accounting for both gaps in the base predictions and repeated choices near the same joint endpoint region. All variants retain six predictions for official evaluation.

Effect of final selection. Table 9 compares final selection with retaining the six refined base candidates from the same pool. Consolidation recovers 302 previously unmatched 8 s futures and loses 9 matches, yielding a net gain of 293 matched scenes. The table reports rates over all valid scenes and over the subset in which a complementary candidate is admitted. Scenes without valid 8 s evaluation in both views are excluded. This comparison isolates the combined selection rule from geometry refinement, but does not separate its individual constraints.

Table 9: Paired change in 8 s joint-future coverage under consolidation. Both subsets contain 302 rescued matches and 9 lost matches. Net gain is reported in percentage points.

Evaluation subset	Scenes	Rescue (%)	Harm (%)	Net gain (pp)
All valid 8 s scenes	41,123	0.7344	0.0219	0.7125
With admitted complements	10,373	2.9114	0.0868	2.8246

B.3 Computational Cost

Table 10 compares JFER with a physically pruned structured base that excludes expansion, refinement, and consolidation. The pruned model was checked for equivalence to the six base trajectories embedded in the full checkpoint. Both configurations return six predictions. The comparison measures the combined cost of the added modules.

Table 10: Module overhead at batch size eight. Time per scene is batch processing time divided by eight, not batch-one response latency.

Configuration	Params (M)	Time/scene (ms)	Peak memory (MiB)
Structured Base	104.385	11.900	831.97
JFER	107.713	17.949	846.34

The additional modules account for 3.19% more parameters and 14.37 MiB more peak allocated memory. At batch size eight, mean processing time per scene increases by 50.83%. These measurements describe aggregate module overhead and do not attribute runtime to individual operations.

Both configurations use an NVIDIA RTX PRO 6000 Blackwell Server Edition with FP32 computation and TF32 enabled. Each setting uses 30 warm-up batches followed by 100 measured batches. Timing excludes data-loader waiting and the official evaluator but includes host-to-device transfers performed inside the forward pass.

C Additional Qualitative Results

This appendix compares the structured base with JFER's final set, examines predictions across horizons, and presents remaining failure cases.

C1: structured base vs. final set. Figure 7 compares the structured base with the final set on four representative scenes. The examples span turning, lane-following, and merging scenes. Paired views show differences in trajectory paths and the distribution of predicted endpoints for the same observed scene.

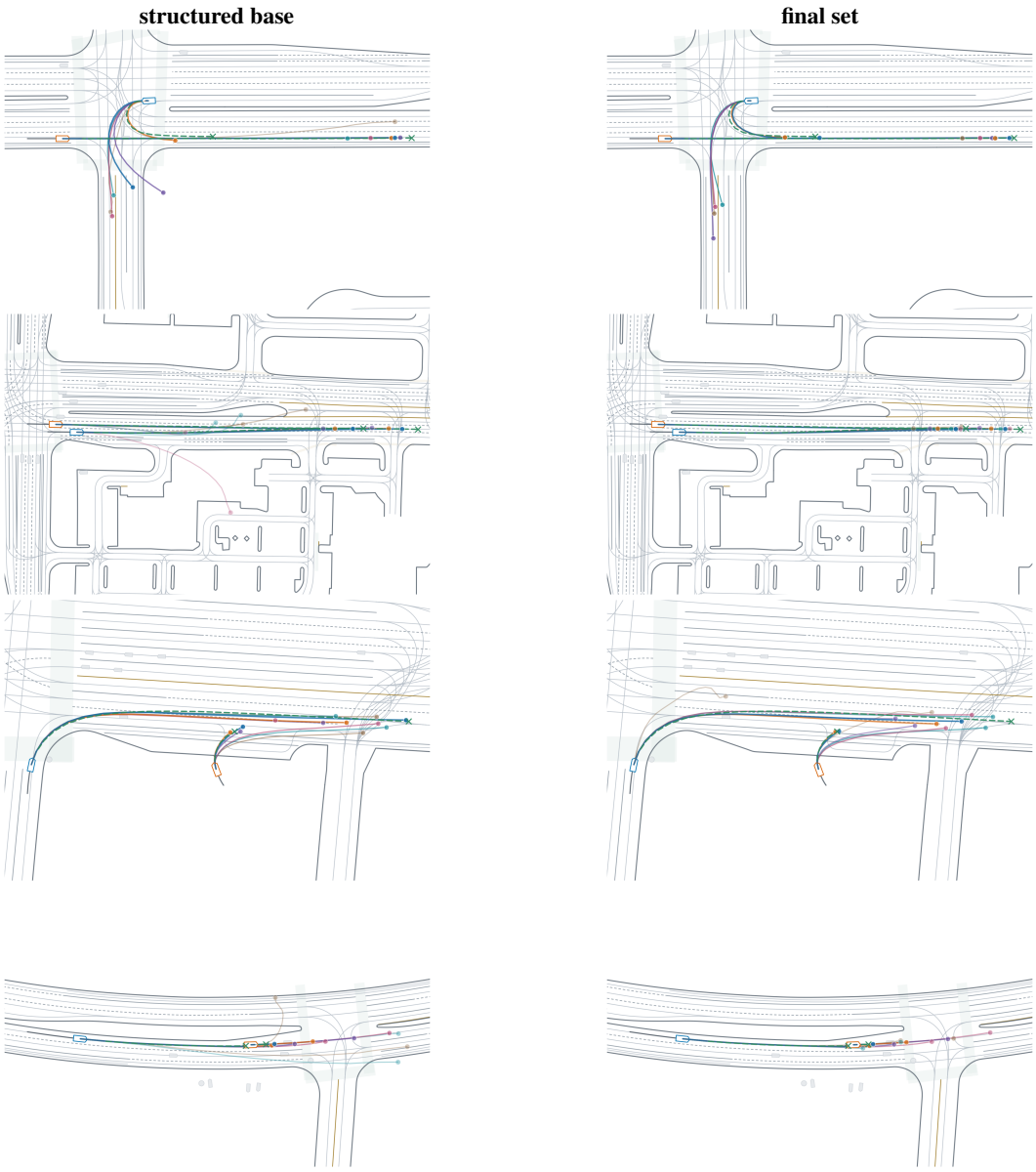


Figure 7: **C1: structured base vs. final set.** Each row shows the same scene, with the structured base on the left and the final set on the right.

C2: Horizon-wise comparison. Figure 8 visualizes the six final joint hypotheses at 3, 5, and 8 s on two scenes. The shorter-horizon predictions remain close to the agents' current motion. At longer horizons, the trajectories separate into alternative paths, showing how the same six predictions represent different future outcomes over time.

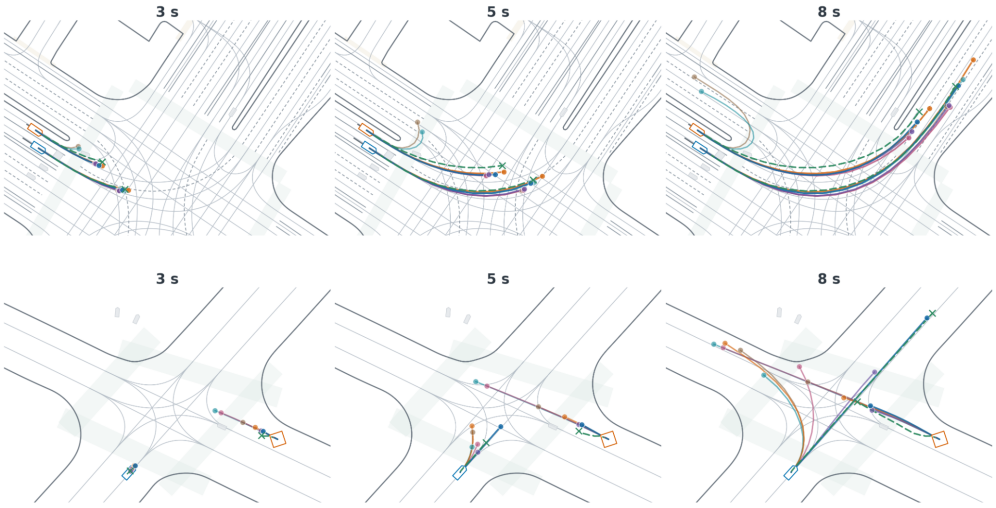


Figure 8: **C2: Horizon-wise comparison.** The final set displayed up to 3, 5, and 8 s in two representative scenes.

C3: Failure cases. Figure 9 presents two remaining errors. The first illustrates an expanded alternative absent from the final set. The second shows inaccurate predictions in a coupled turning interaction. The panels show the resulting trajectories, but do not isolate whether a missing alternative arose from refinement or final selection.

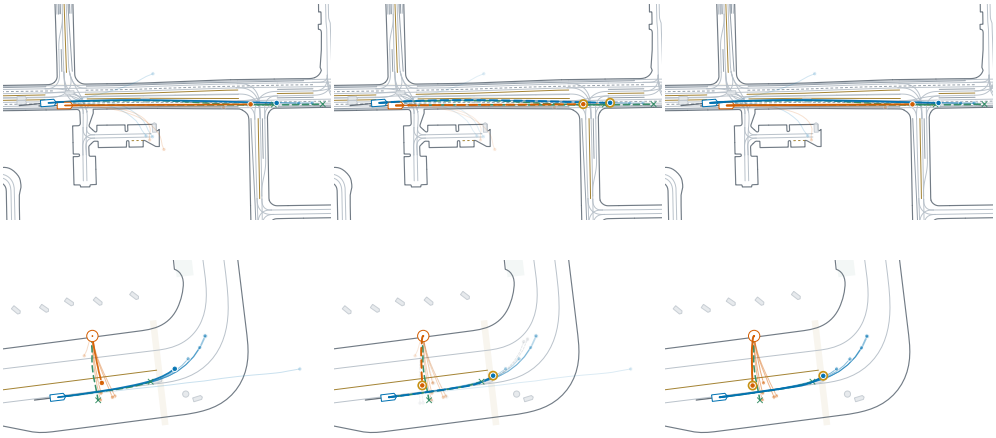


Figure 9: **C3: Failure cases.** Each row shows one remaining failure case: the left column displays the structured base predictions, the middle column displays the expanded pool, and the right column displays the final set after consolidation. The examples show an alternative lost between expansion and the final output, and a difficult turning interaction, respectively.