WHAT SHOULD A VLM TELL AN ACTION EXPERT? COMPARING CONDITIONING INTERFACES FOR LONG-HORIZON FURNITURE ASSEMBLY

Anonymous authors
Paper under double-blind review

ABSTRACT

Translating the semantic understanding, task decomposition, and visual grounding capabilities of vision–language models (VLMs) into continuous robot actions typically requires a conditioning interface that conveys upstream predictions to an action expert. However, systematic evidence remains limited on what information this interface should contain and whether downstream execution can tolerate errors in actual VLM predictions. We use multitask, long-horizon furniture assembly as a diagnostic setting. This domain requires switching across tasks and manipulation stages, localizing parts and their targets, and executing contact-rich skills such as insertion and screwing, thereby amplifying and exposing the downstream effects of interface design and upstream error at the complete-task level. We compare conditioning interfaces constructed from different combinations of three information types: spatial targets, gripper-action cues, and end-effector rotation. Based on this comparison, we recommend the Target–Action Guidance Point (TAGPoint) as an intermediate representation: it encodes a coarse gripper-action cue in the color of a target point, integrating spatial and action information within a unified, image-aligned interface. We further train the action expert with noisy guidance to mitigate performance degradation under inaccurate upstream predictions. Our evaluation covers clean FurnitureBench assembly, guidance-noise training and evaluation, end-to-end execution driven by actual VLM predictions, joint FurnitureBench–AutoMate training, and physical-robot execution. Together, these results provide an empirical basis for selecting measurable and intervenable intermediate representations in VLM–action expert systems.

1 INTRODUCTION

Vision–language models (VLMs) increasingly support instruction understanding, scene reasoning, task decomposition, and visual grounding, making them attractive high-level decision makers for embodied systems (Ahn et al., 2022; Liang et al., 2022; Huang et al., 2023; Brohan et al., 2023). Yet producing a plausible plan is not the same as producing reliable robot motion. The plan must still be converted from an open-ended, discrete output into continuous actions that respect the current scene, embodiment, and contact dynamics. This leaves a central systems question: *what should a VLM tell an action expert?*

One practical approach is to separate open-ended reasoning from continuous control (Figure 1). A VLM interprets the instruction, decomposes the task, and grounds the current interaction, while a Diffusion Transformer (DiT) action expert maps a compact conditioning interface and the current observation to robot actions. Existing systems instantiate this interface at different levels of abstraction. Text plans and programmatic skill calls retain rich semantics but leave grounding and execution to downstream modules (Ahn et al., 2022; Liang et al., 2022); keypoints, waypoints, value maps, and trajectories make spatial intent more explicit but increase the structure that the upstream model must predict (Sundaresan et al., 2023; Huang et al., 2023; Wen et al., 2024). The separation also permits periodic re-anchoring: each guidance refresh predicts from the current observation rather than recursively integrating the previous target or action prediction. This property does not undo failed contacts or eliminate closed-loop error propagation, but it prevents localization residuals from being mechanically accumulated within the interface itself.

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.

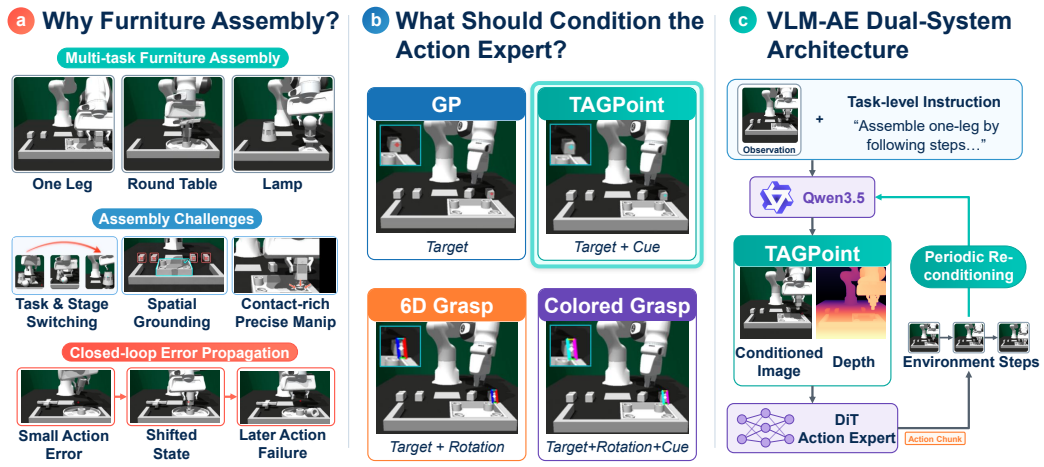


Figure 1: **Task difficulty, conditioning interfaces introduction and system architecture.**

Prior interfaces have established the feasibility of connecting foundation models to robot execution, but they are predominantly evaluated on short-horizon pick-and-place or related tasks. Once the object and destination are selected in such settings, the remaining gap between a correct high-level decision and a successful action can be comparatively narrow. We instead use multitask furniture assembly as a diagnostic setting in which the interface has more ways to matter. A shared policy must switch among furniture, parts, ordered stages, and action modes; localize precise interaction sites; and execute contact-rich placement, insertion, and screwing. A semantic error can invoke the wrong behavior, a spatial error can misalign a precision-critical interaction, and an early execution failure can change the observations and states encountered by all subsequent stages (Heo et al., 2023; Ankile et al., 2024).

We focus on image-space targets because they align directly with visual grounding and allow the action expert to recover metric geometry from its own observation. Within this image-space design, we compare a spatial-only *guidance point* (GP), a *Target–Action Guidance Point* (TAGPoint) whose color encodes whether the current subtask requires an open or closed gripper, a *6D grasp annotation* that augments the projected target with end-effector orientation, and a color-encoded extension of the grasp annotation. These interfaces test the value of providing an explicit 2D spatial target, whether a compact action cue can share the image-aligned carrier of the spatial target and whether richer pose information provides additional benefit. They also raise a complementary question: does the action expert require explicit 3D information to interpret the spatial meaning of a 2D condition? We address this question by evaluating each interface with both RGB and RGB-D observations, thereby measuring how explicit depth affects action generation and the utility of 2D guidance.

Clean scripted guidance isolates the contribution of each interface, but not the semantic errors, spatial offsets, and long-tailed failures of a deployed VLM; the interface must also remain useful as task scale and physical sensing and control change. We therefore ask three questions: (1) what information should condition the action expert; (2) how robust are conditioning interfaces to imperfect guidance; and (3) do the findings extend to larger task sets and physical systems? Across these evaluations, explicit 2D targets improve multitask action generation, coarse action semantics provide further gains at the target, and projected end-effector orientation provides no consistent benefit. TAGPoint remains effective under actual VLM guidance, controlled perturbations, and a larger task set.

Contributions. Our contributions are threefold:

- We construct a controlled RGB/RGB-D observation $\times$ image-aligned guidance comparison for a shared DiT action expert, measuring how 2D position, coarse action semantics, and end-effector orientation affect assembly process, target tracking, and robustness to spatial perturbations.

- We identify TAGPoint as the recommended 2D intermediate representation . It combines an image-aligned spatial target with a compact action cue and remains effective under perturbed guidance and actual VLM predictions.
- We implement and evaluate a complete VLM–TAGPoint–DiT policy pipeline for simulated multitask furniture assembly and test the interface in joint training with a larger set of assembly tasks.

2 Problem Setting and Controlled Study Design

2.1 Multitask Furniture Assembly

FurnitureBench provides a demanding setting for studying conditioning interfaces because successful execution couples long-horizon sequencing with multiskill control, spatial grounding, contact-rich manipulation, and recovery from errors that alter subsequent observations. We use its one-leg, round-table, and lamp tasks, whose dependent stages instantiate five skills: Pick, Place, Push, Insert, and Screw; failure at any critical stage can prevent completion of the assembly (Heo et al., 2023).

Inspection of trajectories labeled successful by the original simulator predicate revealed false-positive completions, motivating three protocol refinements. Each required part pair must remain within its admissible pose window for three consecutive control steps, excluding transient crossings of the target pose. For threaded joints, the final phase about the thread axis is unconstrained, whereas off-axis tilt remains bounded, avoiding an arbitrary axial-phase requirement while rejecting tilted insertions. Finally, we tighten the insertion-direction tolerance for round-table and lamp so that insufficiently seated parts are not counted as stably assembled. All interfaces use the same refined predicate.

The dominant FurnitureBench evaluation protocol trains policies for individual furniture tasks, and even later multitask co-training primarily uses additional assembly tasks as a source of transfer (Heo et al., 2023; Ankile et al., 2024). We instead pool demonstrations from all three tasks and train one shared action expert across furniture, stage orderings, and skills. The policy must therefore infer what action to execute and where to execute it from the current observation and conditioning interface, making the information supplied by the interface directly measurable. At each control step, the action expert maps front- and wrist-view observations, robot state, and an optional conditioning interface to continuous end-effector and gripper actions. Sensor resolution, action parameterization, and execution horizons are specified in the appendix.

2.2 Conditioning Interfaces

Let q denote a conditioning interface. Given observation $o_t = (I_t^{\mathrm{f}}, D_t^{\mathrm{f}}, I_t^{\mathrm{w}}, D_t^{\mathrm{w}}, s_t)$ and a task-specific instruction ℓ_t, an upstream model produces $z_t^{(q)} = h_\phi^{(q)}(o_t, \ell_t)$; controlled comparisons instead provide the corresponding scripted annotation. We distinguish an image-space target $p_t = (u_t, v_t)$, a binary target-site gripper action $m_t \in \{0, 1\}$, and a target end-effector orientation $r_t \in \mathbb{R}^6$. A complete comparison of the information supplied by each interface is provided in Appendix Table 4. RGB and RGB-D without explicit guidance are observation baselines, not additional conditioning interfaces.

TAGPoint encodes m_t in the target point's color, indicating whether the gripper should close or open at that target. For example, Pick closes the gripper around a part, whereas Place opens it to release the part. Pick/Screw share the closing color, and Place/Push/Insert share the opening color. This binds a compact action cue to the image-aligned target. Visual targets and grasp markers are rendered into RGB as $\widetilde{I}_t^{v,q} = \mathcal{R}_q(I_t^v, z_t^{(q)})$.

2.3 Shared DiT Action Expert

Separate ResNet-18 encoders process the front and wrist views. With depth, the first convolution has a zero-initialized fourth channel while retaining the pretrained RGB weights. The encoded views, robot state, and optional low-dimensional condition form the action expert's input:

$$x_t^{(q)} = \left[E_{\mathrm{f}}(\widetilde{I}_t^{\mathrm{f},q}, D_t^{\mathrm{f}}), E_{\mathrm{w}}(\widetilde{I}_t^{\mathrm{w},q}, D_t^{\mathrm{w}}), s_t \right], \qquad \widehat{\mathbf{a}}_t \sim \pi_\theta(\cdot \mid x_t^{(q)}). \tag{1}$$

For RGB-only variants, the depth channels are omitted. The DiT uses self-attention over noisy action tokens; the observation and diffusion timestep modulate its blocks through adaptive layer normalization. We use DDIM at inference; the standard Gaussian noising process and diffusion objective are given in Appendix A. Demonstrations, action-expert capacity, horizons, and optimization are held fixed across interfaces; only the supplied condition changes. Parameterization is detailed in the appendix.

3 What Information Should Condition the Action Expert?

Successful action execution depends on a spatial target, the gripper action, and the end-effector rotation, while a 2D interface leaves metric geometry implicit. The controlled comparisons isolate these information sources and the observation modality used to interpret them. Image-space targets provide the most consistent gains, and a coarse gripper-action cue further improves the more difficult tasks; projected rotation offers no consistent benefit. Explicit depth produces larger gains for conditioned policies by helping recover the 3D meaning of a 2D target.

3.1 Combining Spatial Target and Coarse Gripper-Action Cue Yields the Strongest Interface

To determine what the action expert can exploit from different conditioning interfaces, we compare complete assemblies under accurate scripted guidance. The same DiT action expert is trained on the union of one-leg, round-table, and lamp demonstrations, with the number of demonstrations, model capacity, action horizon, and optimization held fixed. Table 1 compares the no-condition baseline with GP, TAGPoint, 6D grasp annotation, and colored grasp annotation. TAGPoint encodes a coarse gripper-action cue, indicating whether the target-site action requires an open or closed gripper, in the point color. Following the rectangular 2D grasp representation discussed in GraspNet-1Billion (Fang et al., 2020), the 6D grasp annotation projects the target end-effector pose as a gripper rectangle centered at the target; colored grasp annotation adds the same coarse cue to this marker.

Table 1: **Complete-task success across conditioning interfaces and observation modalities.** Values are mean success rate ± sample standard deviation (%) over three training seeds. Each interface is evaluated under clean scripted guidance with RGB and RGB-D observations. Subscripts on RGB-D entries report $\Delta_{\text{depth}} = \text{SR}_{\text{RGB-D}} - \text{SR}_{\text{RGB}}$ in percentage points.

Conditioning Interface	Modality	One-Leg (%)	Round-Table (%)	Lamp (%)	Overall (%)
No Condition	RGB	84.26±4.24	41.67±8.33	18.52±4.24	48.15±3.34
	RGB-D	82.41±1.60 -1.85	49.07±1.60 +7.40	25.93±5.78 +7.41	52.47±1.93 +4.32
GP	RGB	85.42±2.95	45.83±0.00	14.58±2.95	48.61±1.96
	RGB-D	85.19±6.99 -0.23	52.78±15.47 +6.95	33.33±7.35 +18.75	57.10±5.10 +8.49
TAGPoint	RGB	87.50±5.89	45.83±0.00	10.42±2.95	47.92±2.95
	RGB-D	83.33±2.78 -4.17	**58.33±12.11 +12.50**	34.26±1.60 +23.84	**58.64±4.57** +10.73
6D grasp	RGB	85.42±8.84	8.33±0.00	12.50±5.89	35.42±0.98
	RGB-D	**87.04±4.24 +1.62**	14.81±20.85 +6.48	**37.04±5.78 +24.54**	46.30±8.49 **+10.88**
Colored grasp	RGB	85.42±2.95	16.67±11.79	14.58±8.84	38.89±7.86
	RGB-D	**87.04±8.93 +1.62**	22.22±19.25 +5.55	33.33±2.78 +18.75	47.53±5.58 +8.64

With RGB-D observations, the no-condition baseline achieves 82.41%, 49.07%, and 25.93% success on one-leg, round-table, and lamp, respectively, for an overall mean of 52.47%. Adding GP improves the three tasks by 2.78, 3.71, and 7.40 percentage points, raising the overall mean by 4.63 points to 57.10%. The positive difference on every task shows that an explicit image-space target provides broadly useful guidance, with the largest gain on lamp, which has the lowest baseline success.

Adding a coarse gripper-action cue to GP further improves round-table and lamp by 5.55 and 0.93 points, respectively, while reducing one-leg by 1.86 points. TAGPoint consequently reaches 58.64% overall, the highest mean in Table 1. Its gain is concentrated on round-table and lamp, whereas one-leg is already near saturation. Attaching coarse action information to the image-space target can therefore strengthen spatial guidance without transmitting the complete skill identity.

Rotation information does not yield consistent gains across tasks. Relative to GP, 6D grasp annotation improves one-leg and lamp by 1.85 and 3.71 points, but reduces round-table from 52.78% to 14.81%. Colored grasp annotation recovers round-table to 22.22%, which remains well below GP. This task dependence exposes an inherent limitation of a single-view projected rotation mark: when a 3D rotation axis approaches the camera optical axis, its image projection contracts and may overlap other edges, making that orientation component weakly observable or indiscernible. Supplying a 3D rotation therefore does not guarantee that its 2D marker preserves all orientation information. Overall, image-space targets provide consistent gains, while the interface that adds a coarse gripper-action cue achieves the strongest overall performance.

3.2 Depth Helps Policies Interpret 2D Conditions into 3D Actions

An image-space condition identifies an interaction target but does not directly specify its metric geometry. The action expert must combine the 2D guidance with scene structure and robot state to recover the target depth and relative geometry required for continuous control. This raises a central question: is explicit metric scene geometry necessary for the action expert to use a 2D condition effectively, or are RGB visual cues sufficient for its spatial interpretation? We compare RGB and RGB-D observations within each interface family and measure the benefit of explicit geometry as $\Delta_{\text{depth}} = \text{SR}_{\text{RGB-D}} - \text{SR}_{\text{RGB}}$.

Depth improves overall success in all five settings. The depth gains are 4.32 points without an explicit condition and 8.49, 10.73, 10.88, and 8.64 points for GP, TAGPoint, 6D grasp annotation, and colored grasp annotation, respectively. The larger gains for every conditioned policy indicate that depth does more than complement RGB with scene geometry: it also helps ground an image-aligned condition in metric scene geometry, making the guidance more effective for continuous action generation.

The gains are concentrated on the more difficult round-table and lamp tasks. Across interfaces, Δ_{depth} ranges from -4.17 to 1.62 points on one-leg and from 5.55 to 12.50 points on round-table; on lamp, the four conditioned interfaces gain 18.75–24.54 points. Explicit metric scene geometry therefore has little effect on the near-saturated one-leg task but provides larger gains when target depth, insertion direction, and contact alignment are more demanding. Together, these results show that depth geometrically grounds 2D conditions, helping the action expert translate image-space guidance into 3D actions.

4 How Robust Are Conditioning Interfaces to Imperfect Guidance?

Clean scripted guidance excludes the upstream prediction errors that are unavoidable in a VLM–action expert system. We therefore vary guidance noise during training and evaluation, measure complete-task success and fixed-state endpoint behavior, and test the resulting response patterns under actual VLM outputs. Guidance-noise training improves high-noise success while reducing endpoint error and noise-induced displacement. Different training-noise levels produce distinct behavioral responses, while actual VLM guidance preserves the complementary roles of image-space targets and coarse gripper-action cues.

4.1 Evaluation Protocol for Measuring Guidance Robustness

We use TAGPoint as the primary point-based interface and vary spatial guidance noise during both training and evaluation. Let $p_0 \in \mathbb{R}^3$ denote the clean target. The noised target is

$$\xi \sim \mathcal{N}(0, I_3), \qquad \delta_\sigma = \text{clip}(\sigma\xi, -2\sigma, 2\sigma), \qquad p_\sigma = p_0 + \delta_\sigma, \tag{2}$$

where clipping is applied independently along each axis. The perturbation is held fixed within the current skill stage and rendered as a TAGPoint after projection. Nx denotes a per-axis standard deviation of x mm. We compare N0, N6, and N24 during training and evaluate at N0, N3, N6, N12, N24, N48, and N96.

To place actual VLM prediction error on the same controlled noise axis, we define

$$R_{\mathrm{VLM}} = \sqrt{\frac{1}{N}\sum_i \left\|p_i^{\mathrm{VLM}} - p_i^{\mathrm{GT}}\right\|_2^2}, \qquad \sigma_{\mathrm{VLM}} = \mathrm{Interp}(\{(\sigma, R_\sigma)\}, R_{\mathrm{VLM}}), \tag{3}$$

where R_σ is the RMSE obtained by projecting per-axis clipped isotropic Gaussian perturbations under the same 3D target and camera calibration. σ_{VLM} aligns only the projected error magnitude and does not assume that the actual VLM residuals follow a Gaussian distribution. It is used to compare N0, N6, and N24 policies approximately at a controlled scale close to the actual upstream error.

We conduct two complementary evaluations. Complete-task rollouts from the initial states of one-leg, round-table, and lamp measure end-to-end success. Guidance noise, however, changes preceding actions, so policies can enter the same skill stage from different physical states. Complete rollouts therefore cannot isolate the effect of the guidance point itself. We consequently construct a *Fixed-State Intervention Bank* (FSIB) from 18 task–skill stages in successful expert trajectories. States are sampled using a predefined, non-uniform stratification over normalized stage progress, with more frequent sampling near the beginning of each stage. Each record stores the 6D poses of furniture parts and robot and gripper joint states. During evaluation, these states are restored in simulation, velocities are set to zero, and contacts are reconstructed before paired rollouts with different guidance points are executed from the same initial state.

Let x_0 and x_δ denote the endpoints of clean- and noised-guidance rollouts from the same FSIB state. In addition to success rate, we report

$$E_{\mathrm{GT}} = \|x_\delta - p_0\|_2, \qquad \Delta x = \|x_\delta - x_0\|_2, \qquad G = \frac{\langle x_\delta - x_0, \delta\rangle}{\|\delta\|_2}. \tag{4}$$

E_{GT} measures local endpoint accuracy relative to the actual target. Paired rollouts from the same initial state use Δx to capture the behavioral change caused by guidance noise, while G measures the signed component of that change along the perturbation direction.

4.2 Guidance-Noise Training Improves Robustness to Target Perturbations

The N0 policy achieves the highest success rate under clean evaluation, but its performance degrades substantially as evaluation noise increases: success falls from 77.8% under the clean condition to an average of 50.0% over N12–N96, with only 38.9% at the worst noise level. Although N6 and N24 sacrifice 11.1 and 13.9 percentage points of clean performance, respectively, their response curves are markedly flatter. Their mean success rates over N12–N96 reach 60.4% and 62.5%, while their worst-evaluation success rates both increase to 55.6%. Guidance-noise training therefore changes the degradation profile rather than improving only one evaluation level.

The Point VLM has a position-equivalent error of 63.44 mm per axis, between N48 and N96. Although N6 and N24 use substantially smaller training noise, both outperform N0 in this range. Avoiding degradation at the scale of actual upstream errors therefore does not require matching their magnitude exactly; exposure to guidance error is sufficient to reshape the trend.

The FSIB results show that the flatter task-level curves are accompanied by more stable local behavior. Over N12–N96, N6 and N24 reduce E_{GT} by 9.4% and 10.8% relative to N0, respectively, indicating that their endpoints remain closer to the actual target. Their corresponding Δx values decrease by 21.4% and 18.0%, showing that erroneous guidance causes smaller endpoint displacement. Complete-task success, E_{GT}, and Δx thus provide complementary evidence of task-level and behavioral robustness.

4.3 Robustness Reflects Distinct Behavioral Responses across Noise Magnitudes

Lower behavioral displacement under noised guidance can arise from different behavioral responses: a policy may adjust consistently with the guidance direction while the guidance remains reliable, or it may ignore the condition regardless of error magnitude. To distinguish these responses, we examine the signed component G of displacement along the guidance-perturbation direction.

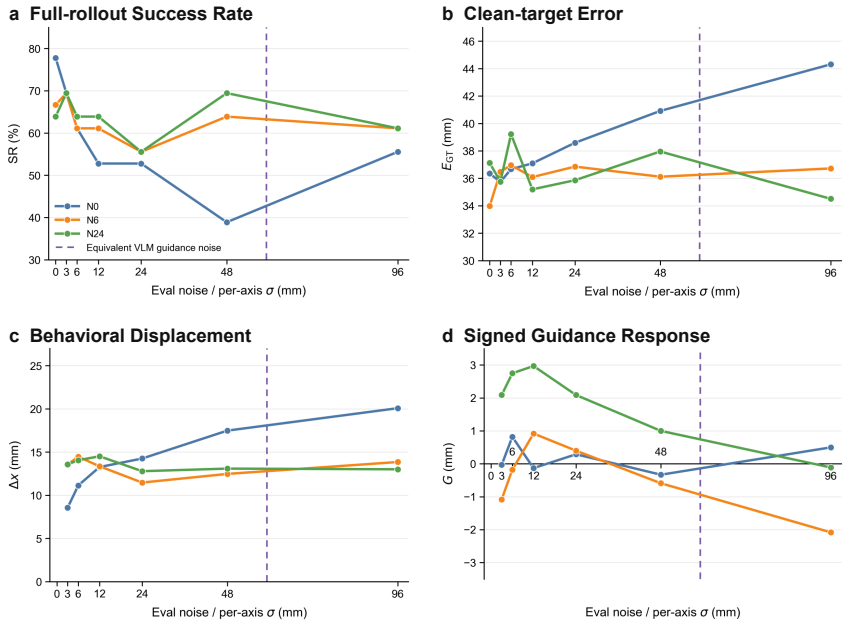


Figure 2: **Guidance-noise training reshapes robustness to noised targets.** TAGPoint policies trained with N0, N6, or N24 target noise are evaluated from N0 to N96, where Nx denotes a per-axis standard deviation of x mm. **a,** Overall complete-task success across one-leg, round-table, and lamp. **b–d,** Fixed-state intervention results for clean-target endpoint error E_{GT}, noise-induced endpoint displacement Δx, and its signed component G along the guidance perturbation.

$G > 0$ indicates that the action endpoint moves towards the noised target, $G < 0$ indicates movement in the opposite direction, and $G \approx 0$ indicates that the perturbation does not produce a stable directional response. Under N3–N96, the N0 policy fluctuates around zero (-0.033–0.082 cm), indicating no stable relationship between endpoint variation and perturbation direction. N6 and N24 instead show the same non-monotonic trend: G increases from small perturbations to N12 and then decreases as noise grows. This trend is consistent with moderate perturbations eliciting measurable target following, followed by reduced directional response once the target deviates substantially from an executable location.

N24 has a larger G than N6 at every evaluation-noise level. For N24, G increases from 0.209 cm at N3 to 0.297 cm at N12 and then falls to -0.011 cm at N96; for N6 it increases from -0.108 to 0.092 cm and then falls to -0.208 cm. Together with N6's lower high-noise Δx, this difference shows that similar task-level robustness can arise through distinct responses: N6 primarily suppresses noise-induced endpoint displacement, whereas N24 retains stronger directional target following while limiting overall displacement.

4.4 VLM-Generated Guidance Connects Controlled Robustness to End-to-End Performance

Controlled Gaussian perturbations isolate the effect of error magnitude but cannot reproduce the task-dependent bias, anisotropy, and non-Gaussian tails of actual VLM errors. We therefore evaluate actual VLM predictions to determine whether the measured response envelope extends to structured upstream errors.

We collect one-leg, round-table, and lamp rollouts with the ResiP expert policy (Ankile et al., 2025) and organize each observation as a VQA sample comprising front and wrist images, a task-specific instruction, and a structured answer. Qwen3.5-2B is fine-tuned using its native response-token cross-entropy. The Point VLM predicts a skill label and 2D target-point coordinates, whereas the Grasp

VLM additionally predicts end-effector orientation as a 6D rotation representation. At each action-chunk replan, the corresponding prediction replaces the scripted interface.

To position the absolute performance of the complete system within existing FurnitureBench research, Table 2 also reports the published JUICER results (Ankile et al., 2024).

Table 2: **End-to-end assembly with actual VLM guidance.** We evaluate shared RGB-D DiT action experts by comparing the no-VLM baseline with interfaces generated by the Point or Grasp VLM. Results from the current study report mean complete-task success ± sample standard deviation (%) over three training seeds.

System / interface	One-leg (%)	Round table (%)	Lamp (%)	Overall (%)
JUICER single-task	74.0	32.0	28.0	44.7
JUICER multi-task	58.0	25.0	12.0	31.7
RGB-D	82.41±1.60	**49.07±1.60**	25.93±5.78	52.47±1.93
RGB-D + GP	**87.04±4.24**	37.04±10.52	34.26±22.62	52.78±12.11
RGB-D + TAGPoint	84.26±1.60	44.44±4.81	**41.67±8.33**	**56.79±2.14**
RGB-D + GP + skill	80.56±4.81	31.48±16.97	26.85±11.23	46.30±8.93
RGB-D + grasp	79.63±8.02	12.96±22.45	34.26±26.40	42.28±18.39
RGB-D + colored grasp	82.41±9.75	15.74±20.48	25.00±14.43	41.05±8.99

Actual VLM residuals exhibit task-dependent bias, directionality, and long tails and therefore cannot be equated with a single magnitude of zero-mean Gaussian perturbation. Nevertheless, these predictions support complete one-leg, round-table, and lamp assemblies, showing that the usable operating range of the action expert does not require perfect upstream localization. TAGPoint achieves the highest overall mean across three training seeds, indicating that the compact spatial–action interface remains usable under structured upstream error.

5 Scalability and Real-World Transfer of Conditioning Interfaces

Task-set scale and physical deployment probe two distinct dimensions of external validity. Joint FurnitureBench–AutoMate training shows that compact spatial targets remain effective across a larger set of assembly tasks, whereas richer cues or rotation markers provide no consistent advantage.

5.1 Scaling to a Larger Set of Assembly Tasks

AutoMate provides large-scale single-step alignment and insertion tasks that complement FurnitureBench's long-horizon assemblies (Tang et al., 2024). We jointly train RGB-D, GP, TAGPoint, grasp, and colored-grasp policies on the three FurnitureBench tasks and demonstrations from 90 AutoMate RL experts, then evaluate the training assemblies (ID90) and ten geometry-clustered held-out assemblies (OOD10). All interfaces use identical trajectories; split construction is detailed in Appendix B.1.

Table 3: **FurnitureBench–AutoMate joint training results across conditioning interfaces.**

Condition	FB Overall	AM ID90	AM OOD10
RGB-D	59.3% +6.79	60.2%	56.7%
RGB-D + GP	**61.1%** +4.01	62.1%	**59.2%**
RGB-D + TAGPoint	**61.1%** +2.47	61.4%	56.7%
RGB-D + grasp	40.7% -5.56	**62.6%**	55.0%
RGB-D + colored grasp	34.3% -13.27	61.4%	55.0%

As the training task set expands, spatial conditions remain usable by the shared action expert, but richer interfaces provide no consistent advantage. On AutoMate ID90, the four spatially conditioned policies achieve 61.4–62.6% success, compared with 60.2% for RGB-D. These modest differences

show that image-space targets remain effective across 90 training assemblies, but do not establish a consistent advantage for action or rotation cues.

On OOD10, GP achieves the highest observed success of 59.2%, exceeding RGB-D by 2.5 percentage points and falling only 2.9 points below its ID90 result. TAGPoint and RGB-D both achieve 56.7%, while both grasp interfaces achieve 55.0%. A compact target therefore remains competitive on held-out assembly identities.

Joint training improves FurnitureBench success for RGB-D, GP, and TAGPoint by 6.79, 4.01, and 2.47 percentage points, respectively, but reduces grasp and colored grasp by 5.56 and 13.27 points. AutoMate is dominated by vertical insertion, where projected rotation axes can occlude one another and remove one rotational dimension. Cross-family transfer therefore depends on compatibility between the condition representation and task distribution rather than on interface richness alone.

5.2 Real-World Evaluation

We additionally train a TAGPoint-conditioned action expert on mixed simulation and physical demonstrations. A successful one-leg rollout confirms that the same image-aligned interface can be consumed within a real sensing–control loop (Appendix C and supplementary video). This qualitative result establishes system feasibility rather than a quantitative real-world success rate.

6 Discussion and Limitations

An image-aligned target can form a compact, intervenable interface between a VLM and continuous control, but the conclusions have three boundaries. First, we evaluate only a DiT action expert. Although the pipeline decouples the upstream condition generator from the downstream policy, the findings remain unverified across broader or more capable control architectures. Second, TAGPoint is designed for gripper-based manipulation and uses color to encode a coarse gripper-action cue. RGB space offers substantial representational capacity, but capacity does not imply learnability; richer cues must be designed and validated for end effectors with different configurations. Finally, the training data and task scale remain limited. Larger and more diverse pretraining data are needed to test whether the observed interface ranking and noise robustness persist.

7 Conclusion

This work systematically examines what conditioning information a VLM should provide to a shared action expert. Explicit 2D targets consistently improve action generation, coarse gripper-action cues further aid difficult tasks, and depth helps recover the 3D meaning of a 2D condition. Training with noisy guidance improves performance under severe perturbations, while actual VLM evaluation confirms that image-space targets and coarse action cues remain effective under structured upstream errors. TAGPoint integrates these signals in a compact, image-aligned representation that remains usable across the expanded AutoMate task set and physical execution. Together, these results provide a measurable basis for selecting intermediate interfaces in VLM–action expert systems.

AI USE STATEMENT

AI-assisted tools were used to organize and edit manuscript prose and to develop the conceptual teaser illustration. Quantitative results come from the recorded evaluations rather than from illustrative generated content. The authors are responsible for verifying the final claims, citations, figures, and text before submission.

ETHICS STATEMENT

Physical furniture assembly can involve unintended contact with the robot, parts, and surroundings. The qualitative physical sequence reported here is not a safety evaluation and does not establish readiness for unsupervised deployment. Quantitative physical evaluation will require a documented safety and intervention protocol.

REPRODUCIBILITY STATEMENT

Section 2 defines the conditioning interfaces and the action-expert input–output mapping. Appendix A reports the resolved architecture, diffusion, optimization, preprocessing, and sequence-training settings used by the registered experiments; Appendix B describes the task composition and AutoMate split. Section 4.1 specifies the guidance-noise construction, fixed-state evaluation, and behavioral metrics, and Appendix E provides the complete train-noise–evaluation-noise grid. Training seeds, resolved configurations, dataset provenance, and checkpoint identifiers are retained in the audited run manifests associated with each reported result.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, and Pulkit Agrawal. JUICER: Data-efficient imitation learning for robotic assembly. *arXiv preprint arXiv:2404.03729*, 2024. URL https://arxiv.org/abs/2404.03729.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement-residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 01–08. IEEE, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11444–11453, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Fang_GraspNet-1Billion_A_Large-Scale_Benchmark_for_General_Object_Grasping_CVPR_2020_paper.html.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2305.12821.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. URL https://arxiv.org/abs/2307.05973.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. *arXiv preprint arXiv:2202.02005*, 2022. URL https://arxiv.org/abs/2202.02005.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022. URL https://arxiv.org/abs/2209.07753.

Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, volume 2026, pp. 90575–90614, 2026.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. *arXiv preprint arXiv:2109.12098*, 2021. URL https://arxiv.org/abs/2109.12098.

Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. KITE: Keypoint-conditioned policies for semantic manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1006–1021. PMLR, 2023. URL https://proceedings.mlr.press/v229/sundaresan23a.html.

Bingjie Tang, Iretiayo Akinola, Jie Xu, Bowen Wen, Ankur Handa, Karl Van Wyk, Dieter Fox, Gaurav S. Sukhatme, Fabio Ramos, and Yashraj Narang. AutoMate: Specialist and generalist assembly policies over diverse geometries. *arXiv preprint arXiv:2407.08028*, 2024. URL https://arxiv.org/abs/2407.08028.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2024. URL https://arxiv.org/abs/2401.00025.

A Additional Implementation Details

A.1 Conditioning Interface Taxonomy

Table 4: **Information supplied by the four conditioning interfaces.**

Interface	2D target	Action semantics	EE orientation
GP	✓	—	—
TAGPoint	✓	✓	—
6D grasp annotation	✓	—	✓
coloredgrasp annotation	✓	✓	✓

A.2 Action-Expert Parameterization

All reported policies use the same diffusion-policy implementation and DiT backbone unless stated otherwise. Tables 5 and 6 list the resolved settings used in the registered runs rather than library defaults.

For the main, robustness, and joint studies, training uses two-GPU DDP with a global batch of 512. Joint training draws equal numbers of FurnitureBench and AutoMate samples in every epoch. Sequence construction uses `pad_after=true` with $T_p - 1$ terminal padding so that every valid observation anchor is retained; padded future actions are removed from the diffusion loss by an action-validity mask. RGB augmentation is applied to both camera views while preserving pixel alignment between RGB, depth, and rendered conditions.

Table 5: **Action-expert architecture.**

Component	Setting
Visual observations	Front and wrist views, 224×224 pixels
RGB encoders	Two independent ResNet-18 encoders trained from scratch; BatchNorm is replaced by GroupNorm
RGB-D encoders	Two independent ImageNet-pretrained ResNet-18 encoders; the RGB kernel is copied to a four-channel first convolution and the depth channel is zero-initialized; depth is normalized with per-camera dataset statistics
Visual projection	128 dimensions per camera view
Robot state	End-effector position and 6D orientation, linear and angular velocity, and gripper width
Action	10 dimensions: 3D end-effector position, 6D orientation, and gripper command
DiT backbone	6 self-attention blocks; width 512; 8 heads; dropout 0.1
Embeddings	256-dimensional sinusoidal diffusion-step embedding; learned action-sequence positional embedding; no RoPE
Temporal horizons	1 observation; 32 predicted actions; first 8 actions executed before replanning
Diffusion	100 training steps; squared-cosine schedule; ϵ prediction with MSE; 16 deterministic DDIM inference steps

Table 6: **Training and preprocessing parameters.** Main denotes the clean interface comparison; robustness denotes guidance-noise training; joint denotes FurnitureBench–AutoMate training.

Parameter	Setting
Optimizer	AdamW, with separate actor and vision-encoder parameter groups
Learning rate	10^{-4} for both actor and vision encoders
Weight decay	10^{-6}
Learning-rate schedule	Cosine decay; 10,000 warm-up updates for both parameter groups
Training length	3,000 epochs; 100 optimizer updates per epoch (300,000 updates)
Global batch size	512 for main, robustness, and joint experiments; 256 for the physical mixed-data policy
Train–validation split	95%–5%
Image augmentation	RGB ColorJitter $(0.3, 0.3, 0.3, 0.3)$ and Gaussian blur (kernel 5; $\sigma \in [0.01, 2.0]$); depth is not photometrically augmented
Normalization	Min–max action and robot-state normalization; RGB uses ImageNet channel statistics; RGB-D additionally uses per-camera depth statistics
Regularization/runtime	No EMA; no mixed precision; gradient clipping disabled
Checkpoint	Final epoch checkpoint for the reported training runs

For an action sequence $\mathbf{a}_t$, all action experts use the standard Gaussian noising process and ϵ-prediction objective

$$\mathbf{a}_t^{(k)} = \sqrt{\bar{\alpha}_k}\mathbf{a}_t + \sqrt{1-\bar{\alpha}_k}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \qquad \mathcal{L}_{\mathrm{DiT}}(\theta) = \mathbb{E}_{\mathcal{D},k,\epsilon}\left\|\epsilon - \epsilon_\theta(\mathbf{a}_t^{(k)}, k, x_t^{(q)})\right\|_2^2. \quad (5)$$

B Dataset and Task Details

The FurnitureBench study uses one-leg, round-table, and lamp demonstrations under the same action-expert training setup for every condition. A task-specific *stage*, such as picking a leg, is assigned one of the five shared *skills*. For controlled comparisons, skill labels and target geometry come from scripted annotations. A 2D guidance point is the calibrated projection of the corresponding 3D target in the same camera frame; rendered markers are policy inputs rather than alterations to the source observation.

B.1 AutoMate ID/OOD Construction

Following the geometric analysis in the original AutoMate study, we sample point clouds from plug and socket surfaces, encode their geometry with a PointNet autoencoder, and select ten cross-cluster representatives as held-out OOD test cases. The remaining 90 assemblies constitute the ID90 training set. The current AutoMate annotations cover Insert, so they do not supply cross-environment examples of all five skills. Per-task demonstration counts, train/validation splits, and the final dataset manifest remain to be inserted after the registered dataset audit.

C Robot Hardware and Control

We reproduce the FurnitureBench task configuration (Heo et al., 2023) and collect 40 physical demonstrations per task through SpaceMouse teleoperation, combining them with 400 simulation demonstrations per task. Both sources use TAGPoint, and the shared action expert is trained on the mixed data. The physical one-leg sequence in Figure 3 uses front-camera inputs and online scripted guidance. It demonstrates use of the same interface under physical visual, depth, control, and contact conditions, but does not establish a real-robot success rate.



Figure 3: **Successful physical rollout with image-aligned TAGPoint guidance.** TAGPoint moves from the manipulated part to its assembly target as the policy uses location and coarse gripper-action cues to complete the task. The complete rollout is provided in the supplementary video.

Planned quantitative evaluation will use RGB and RealSense stereo depth refined by Prompt Depth Anything as policy RGB-D input. Robot and camera calibration, control frequency, action limits, safety supervision, and recovery policy will be specified with the physical trial protocol.

D Related Work

D.1 Data-Efficient Furniture Assembly

FurnitureBench makes long-horizon, contact-rich assembly a reproducible setting for studying imitation and reinforcement learning (Heo et al., 2023). Its sequential stages expose a data-coverage problem: an early error changes the states from which later precision-dependent actions must be generated. JUICER addresses this problem by augmenting bottleneck states with corrective trajectories and expanding the imitation dataset through successful policy rollouts (Ankile et al., 2024). AutoMate instead studies specialist and generalist assembly policies across diverse part geometries (Tang et al., 2024). These works motivate our long-horizon and larger-task-set evaluations. We ask a different, complementary question: with the action-expert architecture and imitation objective held fixed, what information should an upstream system pass to it, and how does error in that information affect complete assembly? Our results are not a protocol-matched performance comparison with those systems.

D.2 Multitask Policy Conditioning

Multitask policies can be specified by language or video demonstrations, as in BC-Z (Jang et al., 2022), by goal images or language in BridgeData V2 (Walke et al., 2023), or by jointly mapping vision and language to actions, as in RT-2 (Brohan et al., 2023). CLIPort explicitly combines semantic and spatial pathways for language-conditioned manipulation (Shridhar et al., 2021). These approaches show that task semantics and spatial information are both valuable, but they make different choices about where the reasoning occurs and what the low-level policy must infer. Diffusion Policy establishes conditional action generation with receding-horizon control (Chi et al., 2023); recent analyses separately examine which components of generative-policy training account for control performance (Pan et al., 2026). Our study holds the action-expert family and training objective fixed and varies the *content* of its condition—skill identity, image-space target, their combination, or target orientation—then tests the same interfaces under controlled and VLM-generated guidance error.

D.3 Keypoints and Visual Prompts for Robot Control

Image-space points can make a spatial decision explicit before continuous control. KITE grounds language to 2D keypoints and executes learned keypoint-conditioned skills (Sundaresan et al., 2023); it is the closest precedent for our GP+skill reference, although its skill-based execution is different from a shared DiT action expert. Any-point Trajectory Modeling uses predicted point trajectories from action-free video to guide policy learning (Wen et al., 2024). We instead study a single target point rendered into the current RGB observation and refreshed by an upstream VLM. The question is not whether points can guide manipulation in general, but which spatial and action semantics should be carried by the point for multitask assembly, and whether that interface remains usable when the upstream prediction is imperfect.

E Complete Noisy-Train/Noisy-Eval Results

Table 7 reports the complete evaluation grid under 0, 6, and 24 mm training noise and 0–96 mm evaluation noise. Full-rollout success rate (SR) measures long-horizon task completion, whereas the matched fixed-state bank isolates the local endpoint response to the perturbed target.

Table 7: **Complete noisy-train/noisy-eval results.** SR is complete-task success over 36 full rollouts per train–evaluation pair (12 per task). The remaining metrics pool 432 short rollouts from 18 stages and 24 fixed expert-restored states per stage. Training noise uses per-axis standard deviations of 0, 6, and 24 mm; evaluation noise N0–N6 uses 0, 3, 6, 12, 24, 48, and 96 mm. SR is reported in percent and endpoint metrics in centimetres. $E_{\mathrm{GT}} = \|x_\delta - p_0\|$ and $E_{\mathrm{input}} = \|x_\delta - p_\delta\|$ measure endpoint error to the clean and corrupted targets, respectively; $\Delta x = \|x_\delta - x_0\|$ measures displacement from the matched clean rollout; and $G = \langle x_\delta - x_0, \delta\rangle / \|\delta\|$ is the signed displacement along the perturbation direction. Bold and underlined values mark the highest and second-highest SR, or the lowest and second-lowest endpoint error or displacement, at each evaluation level; ties share the same formatting. G is not ranked. Δx and G are undefined for N0 evaluation.

Training noise	Evaluation noise	SR (%)↑	$E_{\mathrm{GT}}\downarrow$	$E_{\mathrm{input}}\downarrow$	$\Delta x\downarrow$	G
0 mm	N0 (0 mm)	**77.8**	3.635	3.635	—	—
	N1 (3 mm)	**69.4**	**3.573**	3.645	**0.854**	−0.003
	N2 (6 mm)	61.1	**3.667**	**3.877**	**1.113**	0.082
	N3 (12 mm)	52.8	3.709	4.413	**1.327**	−0.013
	N4 (24 mm)	52.8	3.859	5.908	1.425	0.029
	N5 (48 mm)	38.9	4.091	9.283	1.748	−0.033
	N6 (96 mm)	55.6	4.431	16.383	2.008	0.050
6 mm	N0 (0 mm)	66.7	**3.398**	**3.398**	—	—
	N1 (3 mm)	**69.4**	3.648	3.717	1.355	−0.108
	N2 (6 mm)	61.1	3.696	3.916	1.446	−0.018
	N3 (12 mm)	61.1	3.609	4.275	1.335	0.092
	N4 (24 mm)	**55.6**	3.685	5.679	**1.147**	0.040
	N5 (48 mm)	63.9	**3.612**	**8.790**	**1.247**	−0.059
	N6 (96 mm)	**61.1**	3.672	16.148	1.385	−0.208
24 mm	N0 (0 mm)	63.9	3.712	3.712	—	—
	N1 (3 mm)	**69.4**	3.575	**3.641**	1.357	0.209
	N2 (6 mm)	**63.9**	3.922	4.135	1.406	0.275
	N3 (12 mm)	**63.9**	**3.519**	**4.185**	1.451	0.297
	N4 (24 mm)	**55.6**	**3.586**	**5.600**	1.279	0.209
	N5 (48 mm)	**69.4**	3.795	9.055	1.308	0.100
	N6 (96 mm)	**61.1**	**3.451**	**15.939**	**1.300**	−0.011