# TASTEROUTE: PERSONALIZED ROUTING FOR VIDEO GENERATION

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

## ABSTRACT

Rapid progress in video generation has led to a plethora of models that differ substantially in capability and generation cost. This raises a natural question: can each request be efficiently routed to an appropriate model? We find that even when the consensus of the other annotators is used as an oracle, it agrees with each annotator's own favorite only 34–55% of the time. Motivated by this observation, we introduce TasteRoute, a personalized video-generation router that selects a generator jointly based on the input request, user preferences, and available generation budget. Across text-to-video and image-to-video settings, TasteRoute is competitive with strong simple baselines on preference routing while reducing average generation cost. The cost saving increases under higher budget caps. Finally, we release TasteRoute-3k, a human-annotated dataset containing multi-model video comparisons, quality judgments, preference rankings, and user-profile signals to facilitate future research on personalized and cost-aware video routing.

## 1 INTRODUCTION

Recent advances in conditional video generation have led to models with increasingly high visual quality and controllability (HaCohen et al., 2026; Gao et al., 2025; Team et al., 2025; Google DeepMind, 2025). With just a few clicks, users can now generate videos with remarkable visual fidelity. Nevertheless, in practice, models capable of producing comparably high-quality videos can differ substantially in generation cost. Their strengths may also vary: some may produce finer visual detail, while others may better preserve physical plausibility or achieve a desired style. This raises a natural question: which model should a user choose for a given request?

At first glance, this question may seem ill-posed, as model selection involves many intricate, entangled factors. The choice becomes more manageable, however, when viewed from the user's perspective: does a model produce an output that meets their needs, and which qualities matter most to them? Indeed, if the goal is to generate a short transition shot, clear content and coherent motion may have higher priority than fine-grained visual details. Users may therefore seek an output that meets their needs rather than maximize quality along every dimension(Schwartz et al., 2002). Additionally, individual preferences shape the value of different outputs: one user may favor a stylized rendering, while another prefers photorealism. Even when two models meet the same basic requirements, their outputs may appeal to different users. Model selection should therefore account for both task-specific requirements and individual preferences to determine whether a more expensive model delivers value that justifies its cost.

In this paper, we study the problem of personalized video generation routing. Given a generation request in the form of text and an optional reference image, our goal is to determine which video model to use for a particular user under a given budget. The router makes this choice before generation, considering both whether a model is likely to produce an acceptable output for the input request and how well that output may align with the user's preferences.

The task is far from trivial, in part because learning such a router requires data that captures both sides of this decision. To learn user preferences, we need judgments that preserve individual choices rather than aggregate them into a single consensus ranking. To learn which models suit a given request, we need comparisons of outputs from different generators under the same input. These comparisons must also distinguish basic acceptability from subjective preference, so that clear gen-



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.

eration failures do not obscure differences in taste among otherwise usable outputs. Unfortunately, such data remains scarce due to heavy reliance on human annotations.

To this end, we carefully collect a human-annotated dataset with professional written prompts and videos generated by SOTA models that combines quality assessments, diverse preference rankings, and user profiles for videos generated by multiple models under the same request. We find that annotators agree more often when video comparisons involve unacceptable artifacts, but differ substantially when choosing among acceptable ones. Moreover, the preferred generator varies with both the request and the annotator, confirming that routing must account for both. Using this dataset, we further develop **TasteRoute**, a personalized router that combines representations of the input text and optional image with a learned user embedding. The router predicts basic acceptability alongside a preference score for each candidate generator, allowing it to distinguish shared quality requirements from individual preferences. At inference time, these predictions guide the selection of a generator within the available budget.

Across text-to-video and image-to-video settings, we evaluate TasteRoute against generalized and personalized routing. In generalized routing, TasteRoute achieves performance comparable to a strong kNN baseline. In the personalization setting, TasteRoute-Personal reduces the average generation cost by 8–12% at budget caps of \$1.0–\$2.0 while achieving higher routing accuracy than any other personalized baseline. We further study questionnaire-conditioned routing as a cold-start variant. However, personalization offers limited benefits under tight budgets and remains challenging when users express directly conflicting preferences.

## 2 Problem Formulation

We study video-generation routing, where a router selects a generator for each text-to-video (T2V) or image-to-video (I2V) request under a generation budget. Given a request and a pool of candidate video generators with known generation costs, the router estimates the highest-score of affordable generators. In personalized routing, the selection would additionally include the user's preferences.

Learning such a router, however, would require multi-generator video comparisons under the same request, individual level preference and also separate assessments of video acceptability. Existing video-preference datasets are not well suited for training a video router. They either evaluate earlier models (Liu et al., 2026; He et al., 2026) whose outputs are often dominated by obvious artifacts, or contain relatively few human preference annotations Jiang et al. (2026a). For example, Huang et al. (2026) contains only 1,150 human-annotated pairs, with the remainder labeled by a calibrated Gemini 3.1 judge. Crowd-sourced leaderboards such as Artificial Analysis's Video Arena and Arena AI collect feedback across a broader range of generators (Artificial Analysis, 2026) but do not release their data. Therefore we construct **TasteRoute-3k**, a human-annotated video-routing dataset built around these requirements.

## 3 TasteRoute-3K: Data Collection and Annotation

We begin by collecting 277 video-generation prompts, used as text-only inputs for text-to-video (T2V) and paired with a reference image for image-to-video (I2V). Approximately half of the prompts are drawn from an existing collection (He et al., 2026), while the rest are written by professional video creators with more demanding requirements, such as timestamp-specific motion and camera movement. A prompt may be used in both settings, giving 394 requests (274 T2V and 120 I2V). We then generate candidate videos from a catalog of 30 generator endpoints, covering 16:9 and 9:16 aspect ratios with durations of 4, 6, and 10 seconds, for a total of 3,128 annotated videos. Table 1 in Appendix B summarizes the dataset.

### 3.1 Annotation Pipeline

Nine annotators (four general, five professional video creators) participate in the labeling. Before labeling, all annotators must first complete a calibration meeting on the interface and rubric, covering details such as what a text artifact looks like. We deliberately do not calibrate the aesthetic preference since it is the signal that we wish to capture for personalization.

Three units recur throughout the paper. The *catalog* is the full list of 30 generator endpoints (Appendix G); the *pool* of a request is the generators whose outputs are labeled for it; and a *group* is what one annotator ranks at a time, at most six videos drawn from one pool. A large pool is therefore covered by several overlapping groups; the group is the unit annotators work in, and the pool is the unit of every aggregation and evaluation that follows.

Our annotation pipeline separates video quality acceptability from subjective preference. Annotators first evaluate each video's quality using a shared rubric, then rank candidate videos according to their individual preferences.

**Video Quality and Acceptability** In the first stage, annotators assess each video along four rubric criteria: (i) **Text integrity**: whether visible text is malformed, hallucinated, or unreadable; (ii) **Motion consistency**: whether motion shows smearing, discontinuities, or implausible movement; (iii) **Subject realism**: do subjects have misformed features or physically implausible rendering; and (iv) **Instructions following**: does the video follow the conditioning prompt.

Each criterion is rated *reject*, *borderline*, or *accept*, as is overall acceptability, giving five quality fields per video. The acceptability score remains per-annotator and is never pooled such that a video one annotator rejects may still be ranked by another (Section 4 analyzes these disagreements).

**Human Subjective Preference** Following quality assessment, annotators rank candidate videos according to their individual preferences, with ties allowed. Each group includes two anchor videos shared with the other groups of the same pool, which connect their rankings. We prioritize comparisons between similarly ranked generators to better resolve their relative preferences, while randomly sampling a subset of groups to maintain broader coverage of generator pairs.

**User Preference Profiles** Each annotator also completes a five-item questionnaire covering self-reported preferences for video qualities (e.g., prompt accuracy, natural motion, etc). Because self-reported preferences may not reflect actual choices, each annotator also completes four reference-video comparisons, choosing between videos A and B, indicating no preference, or selecting cannot judge. After selecting either A or B, annotators additionally report their preference strength and up to two perceptual factors (e.g. motion naturalness, camera framing, etc.). These responses provide the cold-start preference signal for video routing (Section 5.1).

### 3.2 General Preference Aggregation

For the general routing task we aggregate annotations within each pool. Each cross-generator video pair takes the outcome that the most annotators chose among win, loss and tie, and carries the number of annotators behind it as its weight; a pair with no strict plurality is ambiguous and is dropped. On the rest we fit a weighted Bradley–Terry (BT) model, one per connected component of each pool's comparison graph, with ties counted as half a win and weights normalized by the number of comparisons the originating listwise ranking induces. Its win probabilities are the soft ranking targets of Section 5.2, yielding 7,062 consensus training pairs. We train on these probabilities because they place every generator of a pool on one smooth scale, but score every analysis and evaluation with the consensus utility of Section 4, which needs no model fit and keeps the ground truth independent of the training target.

## 4 When Does Video Routing Need Personalization?

With this dataset in hand, we examine what a video router needs to learn: (i) how generator strengths depend on the request, (ii) which preferences are shared across annotators, and (iii) when individual preferences diverge. Within a pool, each video pair takes the outcome that the most annotators chose (Section 3.2), and each generator receives a *consensus utility*: its win rate over the pairs it appears in, counting a win as $1$ and a tie as $0.5$, with each pair weighted by the number of annotators behind it. The top-scoring generator is the *consensus winner*.

**No generator wins everywhere, and which one wins depends on the prompt.** If one generator consistently performed best, a fixed choice would suffice. We therefore examine how generator performance varies across prompt attributes (e.g. animation style) in Figure 1 (left): each

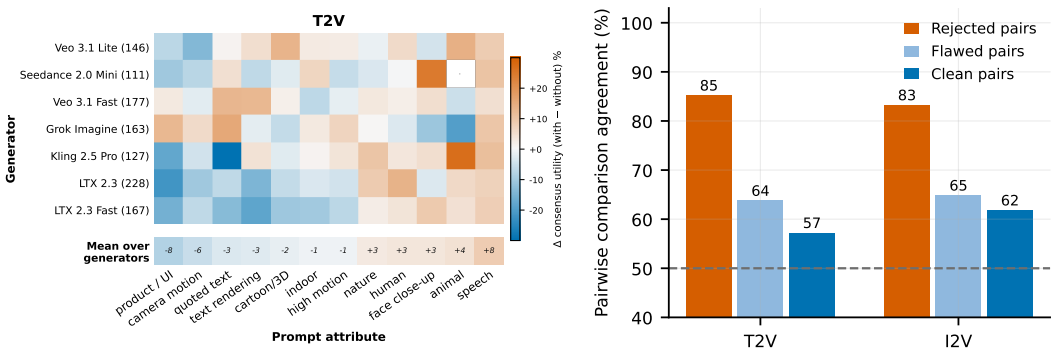



Figure 1: **The best generator changes with the prompt; annotators agree on bad videos but barely on good ones. Left (T2V):** how much better or worse a generator scores on prompts with a given attribute than on prompts without it (%). Different generators lead on different attributes; I2V is in Figure 9. **Right:** how often two annotators who judged the same pair chose the same video, for pairs where one video was rejected, where one carried a defect, and where both were clean.

cell shows the difference between a generator's mean consensus utility on prompts with a given attribute and without the given attribute. In T2V, Veo 3.1 Lite has the highest consensus utility overall, yet it loses 5 of the 12 attributes to three rivals: Seedance 2.0 Mini on face close-ups, Kling 2.5 Pro on animals, and Grok Imagine on product / UI, quoted text and camera motion. A near-zero mean score means the prompt changes *who* wins, which is what a router can exploit.

**Annotators agree on what is unacceptable but not on which is the best.** *Agreement* is how often two annotators who judged the same video pair prefer the same video. We measure it on 19,338 *doubly-judged comparisons* over 376 pools and nine annotators, each one a video pair with two annotators who both expressed a decisive preference. Figure 1(right) splits these pairs by annotator-assessed acceptability and defect flags. When either annotator rejected either video, annotators agree 83–85% of the time: bad output is bad for everyone. Agreement then decreases as the videos get better, to 64–66% when both are acceptable but one carries a defect flag, and to 57% (T2V) and 62% (I2V) when both are clean, barely above the 50% of a coin flip. Quality explains agreement, but prompt content does not: of 16 video attributes, only cartoon or 3D prompts robustly raise agreement on clean pairs ($+18\%$). Prompts that ask for exact on-screen text show the largest negative trend ($-10\%$). See Figure 6 in Appendix B for full breakdown.

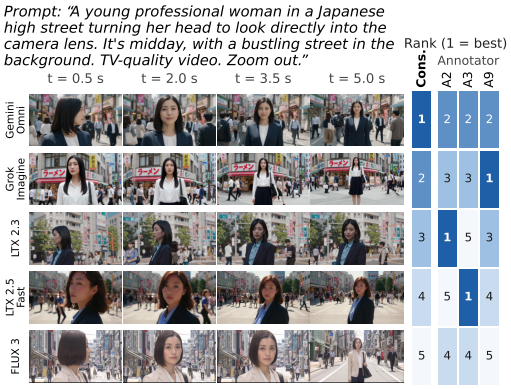



Figure 2: **The consensus winner is nobody's favorite.** One T2V pool in which no video was rejected; the first column is the pool's consensus rank, the rest are the individual annotators'.

**Even when all videos are acceptable, the consensus ranking may not match any annotator's top choice.** Figure 2 showcases a pool in which every video passed every annotator's acceptability check, yet the consensus winner is nobody's first choice: all three rank it second. A harder pool has three favorites in a cycle (Grok $\succ$ FLUX $\succ$ LTX 2.5 Fast $\succ$ Grok), so no aggregation rule can name a winner that more than one annotator endorses (Figure 7, Appendix B). Across pools, 58% have a unique consensus winner, and 7% contain a majority-preference cycle.

**The pool's favorite is the individual's favorite only about half the time.** For every pool an annotator labeled, we take the option preferred by the *other* annotators and check whether it matches this annotator's own top choice; no model training is involved. The consensus favorite matches the annotator's own choice 34–55% of the time, against 19–28% for a random pick. The consensus is a

much better guess than random and still a poor guess for any particular annotator, which is the gap user conditioning has to close. The mismatch is observed across all nine annotators: every annotator agrees with the rest on 63–79% of these comparisons (Figure 8, Appendix B).

These findings motivate a routing framework that captures both shared and individual preferences. A generalized router learns how generator preferences vary across prompts from pooled judgments, while a personalized router additionally models individual preferences among acceptable outputs. A single architecture supports both by conditioning on the prompt and, when available, user information (Section 5.1). The BT soft targets of Section 3.2 represent pooled preferences probabilistically rather than reducing them to hard winner–loser labels.

# 5 TASTEROUTE: PERSONALIZED VIDEO ROUTING

Section 4 split preferences into a shared part (what is acceptable) and a personal part (which acceptable output a user prefers). TasteRoute scores every candidate generator from one frozen embedding of the request, without generating a video, and one architecture yields both a generalized router and per-user routers trained on a user's own feedback. We describe the architecture and how it conditions on a user (Section 5.1), then the objectives of the two variants (Section 5.2).

## 5.1 ARCHITECTURE

Each video generator request is first encoded using the multimodal `Qwen3-VL-Embedding-2B` encoder (Li et al., 2026), taking text prompt for T2V and both text prompt and reference image for I2V. Video generator is represented by a learned embedding. For a user seen in training, we learn $e_u$ along with a shared general-user embedding $\boldsymbol{g}$, which stands in whenever no user information is available (general routing). To prevent overfitting on single individual users, we apply user dropout with probability $p_{\text{drop}} = 0.4$ of replacing user embedding with $\boldsymbol{g}$. For a new user, we instead encode the onboarding questionnaire as text with the same frozen encoder and project it into the user space with a small MLP; this conditioner needs no user-ID table and falls back to $\boldsymbol{g}$ when no questionnaire exists.

All input features are normalized and concatenated, together with their pairwise products, which are then fed to three small MLP heads: a *general* ranking head that sees only the request and generator, a *personalized* ranking head that additionally sees the user vector, and a shared *ordinal quality* head that predicts the four rubric criteria and acceptability of Section 3 as ratings in $\{1, 2, 3\}$. The two ranking heads write scores $s^g$ and $s^p(\cdot, e_u)$, and the quality head writes one latent $q^{(f)}$ per rubric field; throughout, a superscript $g$ marks a quantity computed without user conditioning and $p$ marks one computed with user conditioning. Routing in Section 7 ranks candidates by the ranking heads alone; the quality head is evaluated separately.

## 5.2 TRAINING OBJECTIVES

Every TasteRoute model uses the same two kinds of losses, a pairwise ranking loss and an ordinal quality loss weighted by $\lambda$. The variants differ in what supervises them: which pairs they see, whose preference the target comes from, and which ranking branch is active. Appendix A lists the remaining differences, including the context features.

**Losses.** For candidates $A, B$ of the same request with scores $s_A, s_B$ and a target preference $p_{AB}$,

$$\mathcal{L}_{\text{rank}} = -w\left[p_{AB}\log\sigma\left(\tfrac{s_A - s_B}{\tau}\right) + (1 - p_{AB})\log\sigma\left(\tfrac{s_B - s_A}{\tau}\right)\right], \tag{1}$$

with temperature $\tau$ and the pair weight $w$ of Section 3.2. For the generalized router $p_{AB}$ is the BT win probability aggregated across annotators; for a personalized router it is the annotator's own outcome (1, 0, or 0.5 for a tie). The quality head uses a cumulative-link ordinal loss $\mathcal{L}_{\text{qual}}$: for each annotated rubric field $f$ and threshold $k \in \{1, 2\}$, a binary cross-entropy on $P(Y^{(f)} > k) = \sigma(q^{(f)} - \theta_k^{(f)})$, where $Y^{(f)}$ is the annotator's rating of field $f$ and the $\theta_k^{(f)}$ are learned ordered thresholds, averaged over the fields present.

**TasteRoute-Generic.** One model per data fold, trained on the aggregated pairs $\mathcal{D}_{\mathrm{BT}}$ and the rubric ratings $\mathcal{Q}$ of all annotators through the general head only,

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{rank}}\big(s^g; \mathcal{D}_{\mathrm{BT}}\big) + \lambda\, \mathcal{L}^g_{\mathrm{qual}}(\mathcal{Q}). \tag{2}$$

**TasteRoute-Personal.** One independent model per annotator $u$, trained only on that annotator's pairs $\mathcal{D}_u$ and ratings $\mathcal{Q}_u$. The same pairs train the general head without user information and the personalized head with the (dropout-masked) embedding $\tilde{e}_u$,

$$\mathcal{L}_u = \mathcal{L}_{\mathrm{rank}}\big(s^g; \mathcal{D}_u\big) + \mathcal{L}_{\mathrm{rank}}\big(s^p(\cdot, \tilde{e}_u); \mathcal{D}_u\big) + \lambda\big[\mathcal{L}^g_{\mathrm{qual}}(\mathcal{Q}_u) + \mathcal{L}^p_{\mathrm{qual}}(\mathcal{Q}_u)\big]. \tag{3}$$

At inference it scores with $s^p(\cdot, e_u)$. No annotator's model sees another annotator's labels, so Section 7 measures what one user's own feedback buys. Two TasteRoute baselines reuse these objectives: *TasteRoute-Generic-Pooled* trains the general head on the union of all annotators' individual outcomes without aggregation, and *TasteRoute-Personal-Static* replaces $e_u$ with the questionnaire embedding. All models use $\lambda = 0.25$, selected on the validation split (Appendix A).

## 6 Evaluation Setup

**Evaluation metrics.** For the general video routing, we use five-fold cross-validation with prompt-disjoint splits such that no identical prompt appears in both training and validation or test split. Predictions are evaluated against the consensus utilities mentioned in Section 4. For the personalized video routing, we split each annotator labeling into 15% validation, 15% test with the rest as train. We report the average of three random seed runs.

To study label efficiency, each annotator's training prompt groups are ordered by a diversity criterion seeded with the questionnaire embedding, which serves as the first training unit, and each new group is the one farthest from the centroid of the units chosen so far, and a *cap* $c$ keeps only the first $c$ groups; the matched pooled model at cap $c$ receives $c$ groups in total, balanced across annotators. In both settings, a router chooses among the generators of the pool being evaluated, and held-out labels are never used for routing.

**Metrics and routing under budget.** Utility is the per-pool consensus utility defined in Section 4, computed from individual preferences for personalized evaluation and aggregated across annotators for general evaluation. We report tie-aware Top-1 (the selection is a co-winner), Top-3 (the selection ranks within the top three), Normalized Discounted Cumulative Gain (NDCG). Under budget cap $B$, routers may select only generators priced at or below $B$, but evaluation retains the full pool as ground truth. In particular, NDCG@$n_B$ is normalized against the top $n_B$ generators in the full pool, where $n_B$ is the number of affordable generators. Thus, an unaffordable best generator remains the Top-1 target. We sweep $B \in [\$0.5, \$2.0]$ for both evaluation settings, using a finer grid up to \$1.5 for personalized evaluation.

### 6.1 Baselines

- **kNN** (Li, 2026): retrieves the $k$ most similar training prompts using the same embedding models as TasteRoute and ranks generators by the mean observed utility ($k$ chosen on validation); *Same-user kNN* implements the personalized version of kNN routing by limiting to only a single annotator rather than the aggregated mean observed utility in kNN.
- **Octo** (Jiang et al., 2026b): an agentic video-adapted image router. It selects the cheapest generator whose remembered pass rate for the prompt's capability bucket exceeds a threshold, then verifies the output, and escalates on rejection for up to four attempts. We evaluate VideoScore2 (He et al., 2026), GPT-5.6, Gemini 3.7 Flash as verifiers, and include human labels as signal for upper-bound (*Octo-Human-Label*).
- **FrugalGPT** (Chen et al., 2024a): A video-routing variant of FrugalGPT routes each input through a cascade of video models, first generating a video and then evaluating its quality with a video-quality scorer; if the output is deemed sufficiently reliable, the process stops, otherwise the input is escalated to a more capable, higher-cost model.
- **Hierarchical User-BT**: first fits a shared generator ranking from all pairwise votes, then it learns a small per-annotator correction score based on each annotator's history preference. The adjustment weight $\lambda$ chosen based on validation set.

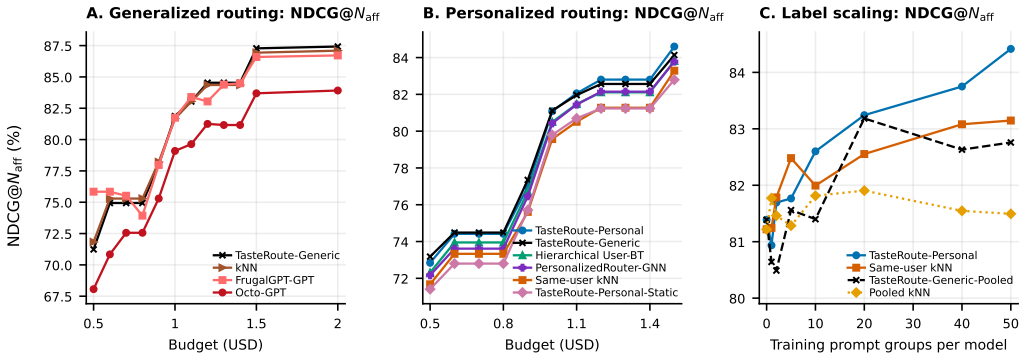



Figure 3: **Routing quality by limiting the cost available to the router in two settings**: generalized routing (A) and personalized routing (B). In both settings, TasteRoute (Generic, Personal) performs competitively against the strong kNN baseline. (C) We also compare training dataset scaling across all routers and find that TasteRoute-Personal scales better as data continues to grow. All performance is measured using NDCG.

- **PersonalizedRouter-GNN** (Dai et al., 2025): We adopted graph-based personalized router with video generators as nodes, inputs as query nodes, and user-conditioned link prediction.
- **TasteRoute variants** (Section 5.2): *TasteRoute-Generic-Pooled* and *TasteRoute-Personal-Static* for cold start scenario where no user routing preference was recorded.
- **Random**: uniformly selects one generator from the valid candidate set for each input.

The details of implementation can be found in Appendix A.

## 7 Performance of Generalized and Personalized Routing

As Section 4 separates video routing into consensus and personalized layers, in this section, we train consensus preferences alone for generalized (user-agnostic) routing, and then include annotator preferences to improve personalized routing. We also study the needed data amount per-annotator to trade off labeling effort against routing quality .

### 7.1 Generalized Routing Results

Figure 3 reports NDCG for this subsection and the next: panel A grades generalized routing, panels B and C personalized routing. Top-1 and Top-3 give the same picture and are in Appendix C.

**Learned routing offers modest gains over retrieval.** Figure 3A shows that TasteRoute-Generic and kNN achieve similar NDCG across budget caps. At the $2.0 cap, TasteRoute-Generic reaches 42.0% Top-1, compared with 39.7% for kNN, 38.2% for FrugalGPT, and 33.6% for Octo. These results suggest that learning prompt–generator interactions offers limited improvement over retrieval under pooled preferences. With an oracle verifier using held-out human labels, Octo reaches 58.0% Top-1, highlighting the role of verification quality in generate-then-verify routing.

**Prompt attributes alone are insufficient for generator selection.** Next we examine whether a simpler attribute-based lookup can replace a learned router. Figure 10 in Appendix B shows that generator preferences vary across prompt attributes, with TasteRoute-Generic's Top-1 ranging from 48.6% on cartoon and 3D prompts to 30.4% on prompts requesting rendered text. Prompt attributes provide some useful routing signal, yet a per-attribute lookup table achieves only 37.0% Top-1 accuracy, compared with 36.6% for a single global-best generator and 39.9% for TasteRoute-Generic. This suggests that the learned router exploits prompt-dependent information beyond coarse attribute categories. To examine its sensitivity to request, we modify one attribute phrase while keeping the remaining prompt unchanged. These edits alter TasteRoute-Generic's top-ranked generator in 5–55% of cases across ten attributes, with an average flip rate of 25%.

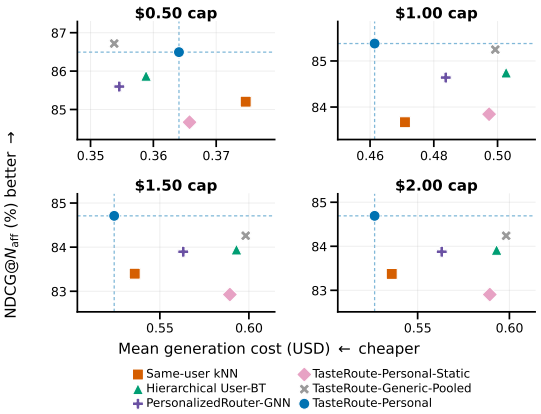



Figure 4: **Cost against routing quality.** Upper left is cheaper and better; dashed lines mark the personalized router. Quality here is NDCG over the affordable pool, not over the full pool of Figure 3.

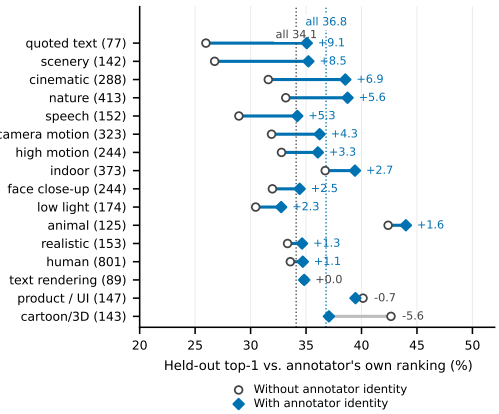



Figure 5: **Personalization by prompt attribute.** One router scored with and without the annotator's identity, graded against that annotator's own ranking. The bar is the gain; dotted lines are the two overall rates.

### 7.2 Personalized Routing Results

**Personalization becomes more useful as the budget expands.** Figure 3B shows that both TasteRoute variants perform competitively with kNN, Hierarchical User-BT, and PersonalizedRouter-GNN across budget caps. TasteRoute-Personal performs similarly to TasteRoute-Generic below $1.1 per video but pulls ahead as the budget increases. At lower caps, the limited choice of affordable generators leaves less room for individual preferences to influence routing. For example, an annotator's preferred generator is unaffordable in approximately 20% of pools at $0.5, compared with almost none at $1.5. As more generators become affordable, TasteRoute-Personal can better exploit differences between individual and pooled preferences.

**How much user feedback is needed for personalization?** To study personalization with limited user feedback, we vary the number of labeled prompt groups used for training (Figure 3C). At each training size $c$, TasteRoute-Personal uses $c$ groups from one annotator, while TasteRoute-Generic-Pooled uses the same number of groups across annotators. With fewer than 10 groups, the Generalized router and Same-user kNN perform better. TasteRoute-Personal surpasses the pooled router at approximately 10 groups and Same-user kNN by 50. Its NDCG continues to improve at the largest training size, suggesting that additional user feedback may further improve personalization.

## 8 Analysis on Personalized Router

Building on the overall performance gains, we further analyze how personalized routing is improved from generic routing in terms of generation cost and alignment with individual preferences.

### 8.1 Trade-offs between Cost and Preference

**The preferred generators are also the cheaper ones.** Figure 4 plots each router's mean cost against routing quality at every budget cap. NDCG of routing quality is calculated over the affordable pool instead of the full pool used in Figure 3. At the $1.0, $1.5, and $2.0 budget caps, TasteRoute-Personal is the cheapest and most accurate router, reducing cost by 8% to 12.0% compared to the most expensive alternatives at the same caps. Only at the $0.5 budget cap, TasteRoute-Generic-Pooled is marginally ahead on both axes. TasteRoute is trained without cost supervision, and the budget cap is only an affordability filter at inference time, so this saving is a byproduct of the learned preference ranking: the preferred generators are not the expensive ones.

**Learning on cost harms accuracy.** Saving cost is also learnable: following Li (2026), we applied a user-defined trade-off parameter governing the preference for performance versus cost, the full sweep over parameters and budget caps is given in Table 3 (Appendix D). A parameter small enough makes Top-1 unchanged at every cap and saves at most 0.03%. Larger parameters trade accuracy for cost: at the \$1.5 cap a parameter value of $0.3$ reduces cost by 9.8% for 1.4 points of Top-1, and $1.0$ by 25.2% for 2.8 points. This gives us an operating point to trade-off between explicit accuracies and generation costs.

### 8.2 Understanding Personalization Gains

We first test whether personalization captures annotator-specific preferences using held-out video pairs on which two annotators express opposite preferences. TasteRoute-Generic makes the same prediction for both annotators and achieves 50% accuracy by construction. TasteRoute-Personal reaches 55.2%, compared with 52.4% for Same-user kNN and 52.2% for PersonalizedRouter-GNN, indicating modest sensitivity to individual preferences (Figure 14, Appendix C). We then examine whether this sensitivity translates into larger routing gains when annotators disagree.

**Personalization gains are larger when annotators share a preferred generator.** We partition 120 matched annotator–pool cohorts by whether any annotators share a top-ranked generator. Among the 54 cohorts with a shared winner, personalization improves Top-1 from 36.7% to 50.1%; among the remaining 66 cohorts, it improves Top-1 from 25.9% to 28.7%, with the latter improvement not statistically conclusive. Thus, although personalization distinguishes some conflicting pairwise preferences, its larger routing gains occur where preferred generators overlap.

**How do personalization gains vary across prompt attributes?** In a separate held-out T2V analysis, we compare the general and annotator-conditioned heads of the same jointly trained router against each annotator's own ranking (Figure 5). Personalization raises overall Top-1 from 34.1% to 36.8%. Gains range from $-5.6$ percentage points for cartoon/3D and $-0.7$ for product/UI to $+8.5$ for scenery and $+9.1$ for quoted text. The largest gains occur in the two groups with the lowest general-head accuracy, highlighting personalization's value where general routing is weakest.

## 9 Conclusion

We presented TasteRoute-3k, a human-annotated dataset of multi-generator video comparisons, and TasteRoute, a personalized video-generation router that selects a generator from the request, the user's preferences and the available budget. TasteRoute-3k shows that preferences over video generators are both diverse and conflicting across annotators, and that a router conditioned on annotator identity improves over generalized routing once the budget no longer constrains the choice. TasteRoute outperforms every personalized baseline at every budget cap and, from the \$1.0 cap upward, is simultaneously the cheapest and the most accurate router. Our analyses of opposite-preference pairs, of the cost–preference trade-off, and of the link between prompt attributes and personalization show that the router encodes annotator-specific preference but mostly corrects the consensus rather than resolving conflicts between users. Closing that gap, with user-conditioned features that raise accuracy without raising generation cost, is the direction we see for future work, alongside understanding how human preference over generated video differs from preference over text and images.

## 10 Limitation

The gap between the best router and the oracle remains large (42.0% against 58.0% Top-1 at the \$2.0 cap), and widens as more generators and larger budgets make preferences harder to predict. Beyond this, our personalized router mostly corrects the consensus rather than resolving conflicts between annotators (55.2% on opposite-preference pairs), results rest on nine annotators and a fixed generator catalog, and the observed cost saving reflects these annotators' preference for cheaper generators rather than a general property.

### AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting, literature searches, and partial experiment implementation. The authors reviewed the AI-assisted content and take responsibility for the paper's claims, results, and materials.

### ETHICS STATEMENT

TasteRoute-3k incorporates prompts from an existing dataset (He et al., 2026), supplemented with prompts written by professional video creators. Reference images for image-to-video generation were generated using FLUX 3 and Gemini 3.1 Flash Image. We follow the applicable usage terms of the source datasets and model providers.

Although personalized routing can reduce video generation costs, it may also lower the barrier to producing harmful or misleading synthetic content. TasteRoute does not introduce an independent content-safety mechanism and relies on the safeguards of the underlying video generators, which may differ across models and do not eliminate potential misuse.

### REPRODUCIBILITY STATEMENT

The repository contains training and evaluation code, data preparation scripts, and per-example results. The scripts construct training data from TasteRoute-3k. Appendices A contain detail the model architecture, training settings, evaluation instructions, and metric conventions.

## REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7120–7129. IEEE, 2017.

Artificial Analysis. Video generation arena leaderboard. `https://artificialanalysis.ai/video/arena`, 2026. URL `https://artificialanalysis.ai/video/arena`. Accessed: August 2026.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a. URL `https://openreview.net/forum?id=cSimKw5p6R`.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL `https://openreview.net/forum?id=7RQvjayHrM`.

Zhongjie Dai, Tao Feng, and Jiaxuan You. Personalizedrouter: Personalized llm routing via graph-based user preference modeling. 2025.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/b47d93c99fa22ac0b377578af0a1f63a-Paper-Conference.pdf`.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for LLM selections. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=eU39PDsZtT`.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Google DeepMind. Veo: A text-to-video generation system. Technical report, Google DeepMind, May 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf. Veo 3 Technical Report.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, Keming Wu, Benjamin Schneider, Quy Duc Do, Zhuofeng Li, Yiming Jia, Yuxuan Zhang, Guo Cheng, Haozhe Wang, Wangchunshu Zhou, Qunshu Lin, Yuanxing Zhang, Ge Zhang, Wenhao Huang, and Wenhu Chen. Videoscore2: Think before you score in generated video evaluation. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=MpkVh4jH44.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

Xiaoyan Hu, Ho-Fung Leung, and Farzan Farnia. PAK-UCB contextual bandit: An online learning approach to prompt-aware selection of generative models and LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Yinming Huang, Shuyuan Tu, Xi Yan, Zihan Yang, Jianhua Han, Xu Hang, Yu-Gang Jiang, and Zuxuan Wu. Va-judger: Reward modeling from human preference feedback for joint video-audio generation. *arXiv preprint arXiv:2608.18607*, 2026.

Longteng Jiang, DanDan Zheng, Qianqian Qiao, Heng Huang, Huaye Wang, Yihang Bo, Bao Peng, Jingdong Chen, Jun Zhou, and Xin Jin. Vga-bench: A unified benchmark and multi-model framework for video aesthetics and generation quality evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Xu Jiang, Bin Chen, Gehui Li, Yule Duan, Ronggang Wang, and Jian Zhang. Octot2i: A self-evolving agentic text-to-image router. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Esma Kilic and Ethem Alpaydin. Learning the areas of expertise of classifiers in an ensemble. *Procedia Computer Science*, 3:74–82, 2011.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Chen Keqin, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Yang Li. Rethinking predictive modeling for llm routing: When simple knn beats complex learned routers, 2026. URL https://arxiv.org/abs/2505.12601.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems*, 2026.

Deepak Babu Piskala, Vijay Raajaa, Sachin Mishra, and Bruno Bozza. Optiroute dynamic llm routing and selection based on user preferences: Balancing performance, cost, and ethics. *International Journal of Computer Applications*, 2024. URL http://dx.doi.org/10.5120/ijca2024924172.

Parham Rezaei, Farzan Farnia, and Cheuk Ting Li. Be more diverse than the most diverse: Optimal mixtures of generative models via mixture-ucb bandit algorithms. In *International Conference on Learning Representations*, 2025.

Barry Schwartz, Andrew Ward, John Monterosso, Sonja Lyubomirsky, Katherine White, and Darrin R Lehman. Maximizing versus satisficing: happiness is a matter of choice. *Journal of personality and social psychology*, 2002.

Yu-Chien Tang, Jun-Chen Hung, Wen-Chih Peng, and An-Zi Yen. Vdar-router: Adaptive llms routing via verbalized query difficulty analysis retrieval. *arXiv preprint arXiv:2607.18098*, 2026.

Kling Team, Jialu Chen, Yuanzheng Ci, Xiangyu Du, Zipeng Feng, Kun Gai, Sainan Guo, Feng Han, Jingbin He, Kang He, et al. Kling-omni technical report. *arXiv preprint arXiv:2512.16776*, 2025.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching LLMs multi-round routing and aggregation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DWf4vroKWJ.

# APPENDIX

**A Implementation Details** **14**

**B Additional Dataset Analysis** **14**

**C Additional Routing Results** **17**

**D Cost-Supervision Trade-off** **18**

**E Related Works** **19**

E.1 Model Routing . . . . . 19

E.2 Model Routing In The Age of Generative Model . . . . . 20

**F Showcase of Labeling UI** **20**

**G Video Generators and Cost** **22**

# A Implementation Details

**Encoder and context features.** Requests are embedded once by the frozen `Qwen3-VL-Embedding-2B` encoder (2048-D, prefixed with the task name; I2V requests include the reference image). The generalized five-fold routers project the context to 32 dimensions with a PCA fitted on each fold's training set; the personalized routers use the raw embedding. Generator identity is a family embedding plus an endpoint-specific residual; candidate features are the normalized context, generator and user vectors together with their three pairwise products.

**Training.** AdamW with weight decay $0.01$, batch size 64 for pairs and 32 for rubric rows, temperature $\tau = 1$, user dropout $0.4$, up to 50 epochs with early stopping (patience 10). The general and the personalized ranking stream are iterated in lockstep, so the number of ranking updates does not depend on the quality weight. Generalized routers: learning rate selected per fold from $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}\}$ by inner-validation Top-1 then NDCG. Personalized routers and the pooled baseline: learning rate $3 \times 10^{-4}$, checkpoint with the lowest validation ranking loss (both ranking streams weighted equally for the personalized routers; the general stream only for the pooled baseline). Seeds 13, 29 and 47.

**Quality weight.** $\lambda$ was swept over $\{0, 10^{-4}, 10^{-3}, 0.25\}$ with the ranking step budget held fixed across $\lambda$. Any positive weight trains the quality head (test quality AUC $0.79$ for the generalized router versus $0.51$ untrained) without a measurable change in ranking accuracy; $\lambda = 0.25$ gave the best validation Top-1 and is used for every model. Quality prediction is claimed for the generalized router only; the personalized routers keep the term for a uniform objective.

**Personalization protocol.** For each seed, 15% of prompt groups are held out for validation and 15% for test, identically for all nine annotators. The support order of an annotator is *centroid-diverse*: starting from the questionnaire embedding, the next prompt group is the one whose centroid embedding is least similar to the centroid of the groups already selected. A cap $c$ keeps the first $c$ groups; the matched pooled model at cap $c$ takes $c$ groups in total in balanced round-robin order over annotators. Reported curves use caps $\{1, 2, 5, 10, 20, 40, 50\}$ and the full data. Same-user kNN uses $k = 20$.

**Agentic baselines.** Octo keeps a capability memory of per-generator pass rates over prompt-derived capability buckets built from training-fold verifier outcomes, routes to the cheapest generator whose expected pass rate exceeds a threshold (threshold and minimum bucket count selected on validation), and escalates to the next-cheapest generator for up to four attempts when the verifier rejects the output. The verifier rates visual quality, prompt alignment and physical consistency on a 1–5 scale and accepts a video when the mean is at least $4.0$ and every dimension at least $3.0$. FrugalGPT learns, per budget, a cascade order of up to four generators and per-stage acceptance thresholds on the training split, selects the configuration on validation, and replays it frozen at test time under a per-pool cumulative cost cap; the last affordable stage always returns its output. Both use the verifier's mean rubric score as the answer scorer; generation cost accumulates over attempts and verifier API or compute cost is reported separately. kNN was tuned over $k \in \{1, 10, 20, 30, 50, 100\}$.

**Costs.** Generation cost is the generator's catalog price per video at the time of collection, known before generation; prices are frozen for all experiments.

# B Additional Dataset Analysis

This appendix collects the dataset analyses that Section 4 summarizes in one or two sentences. Every estimator, seed and input is the same as for the corresponding figure in the main text.

**Which generators win, and how often.** Table 2 lists the five most frequent pool-level winners. The distribution is flat: the leader takes 10.1% of the pools, nine of the 30 generators each take more than 5%, and 28 win at least one pool, which is why a single default generator is a weak policy and routing has something to exploit.

Table 1: TasteRoute-3K at a glance. A *pool* is one request in one setting together with every generator labeled for it; a *group* is the unit one annotator ranks at a time. A *comparison* is one ordered pair of videos within a group; the consensus outcome of a pair is the label chosen by the most annotators, and pairs with no strict plurality are *ambiguous* and excluded from the consensus targets.

| Quantity | Count | Definition |
|---|---|---|
| Distinct prompts | 277 | unique prompt texts |
| Pools | 394 | one per request, i.e. prompt × setting (274 T2V, 120 I2V); 393 evaluable |
| Groups | 812 | 2.06 per pool, at most six videos each |
| Annotated videos | 3,128 | distinct (pool, generator) generations |
| Generator endpoints | 30 | catalog of Appendix G; 28 win at least one pool |
| Annotators | 9 | four general annotators, five professional video creators |
| Consensus pairs | 8,181 | distinct video pairs with an aggregated outcome |
| non-ambiguous | 7,062 | training targets after dropping ambiguous pairs |
| Doubly-judged comparisons | 19,338 | (comparison, annotator pair) with two decisive votes |

**Prompt attributes do not explain who agrees with whom.** Figure 6 gives the full version of the attribute effect quoted in Section 4: for each of the 16 keyword-detected attributes, the change in preference agreement among clean pairs when the attribute is present. Only cartoon or 3D prompts have an interval that excludes zero, so prompt content explains little of the disagreement that remains once both videos are acceptable.

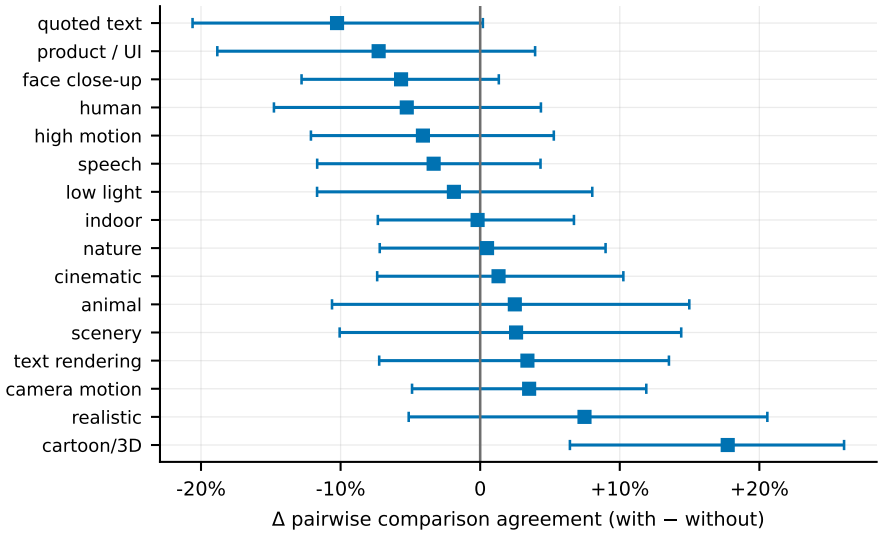



Figure 6: **Prompt attributes barely move agreement.** Among clean pairs (T2V and I2V pooled), the change in preference agreement when a keyword-detected prompt attribute is present versus absent, with pool-level bootstrap 95% intervals, since comparisons within a pool are correlated. Only cartoon or 3D prompts have an interval that excludes zero.

**A pool with no possible consensus winner.** Figure 7 is the second of the two qualitative examples in Section 4. Three annotators accept every video and each names a different favorite, and their majority relation is cyclic, so no aggregation rule can produce a winner that more than one of them endorses. Cycles of this kind occur in 7% of all pools.

**How much a consensus-only router can satisfy each annotator.** Figure 8 is the evidence behind the 34–55% figure quoted in Section 4 and in the abstract. Panel A takes, for each annotator and pool, the favorite of the *other* annotators and asks whether it is also that annotator's own top choice;

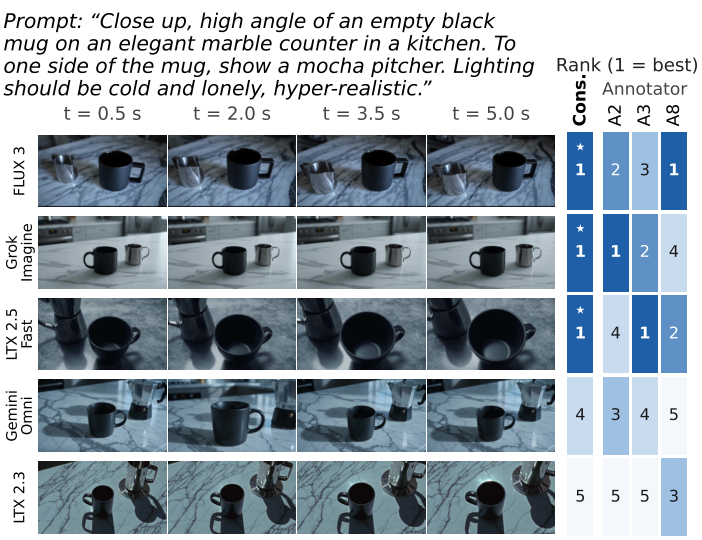



Figure 7: **Cyclic favorites.** A T2V pool in which no annotator rated any video Not acceptable, laid out as in Figure 2, where ⋆ marks a rank shared by more than one generator. A2 prefers Grok Imagine, A3 LTX 2.5 Fast, and A8 FLUX 3; the majority relation is cyclic (Grok ≻ FLUX ≻ LTX 2.5 Fast ≻ Grok), and the consensus ties all three for first.

panel B repeats the question one comparison at a time. The narrow spread across annotators rules out a single unusual labeller as the cause.

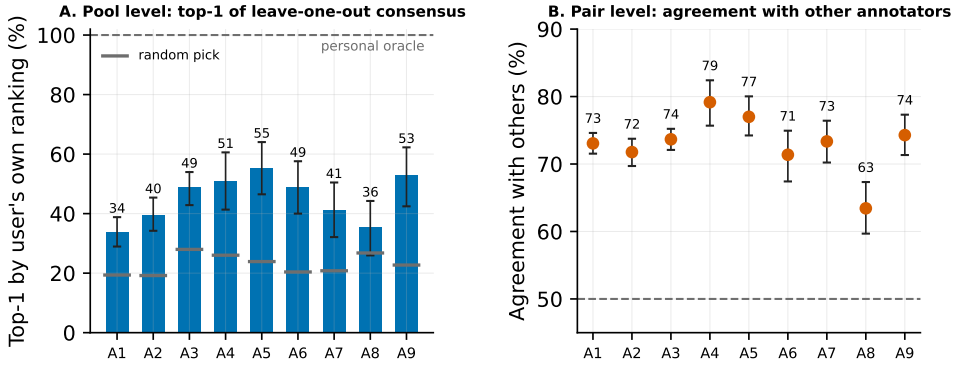



Figure 8: **How far consensus gets each annotator.** (A) Pool level: how often the favorite of the *other* annotators on the same pool is also this annotator's own top choice (tie-aware). Gray marks a random pick; the dashed line at 100% (*personal oracle*) is a router that always matches the annotator. (B) Pair level: how often the annotator prefers the same video as another annotator in a single comparison. Bars in both panels are pool-level bootstrap 95% intervals.

**Prompt-dependent generator strength in image-to-video.** Figure 9 is the I2V counterpart of Figure 1. The winner changes on 8 of the 12 attributes, three more than in T2V, so the prompt-dependence that motivates routing is not specific to the text-to-video setting.

**The generalized router against prompt-independent references.** Figure 10 is the panel that Section 7 refers to when it reports 39.9% for the router against 37.0% for an attribute lookup table and 36.6% for a single global best. Per attribute the router ranges from 48.6% to 30.4%, but the lookup table tracks it only loosely, which is the evidence that the gain comes from the context embedding rather than from the keyword attributes.

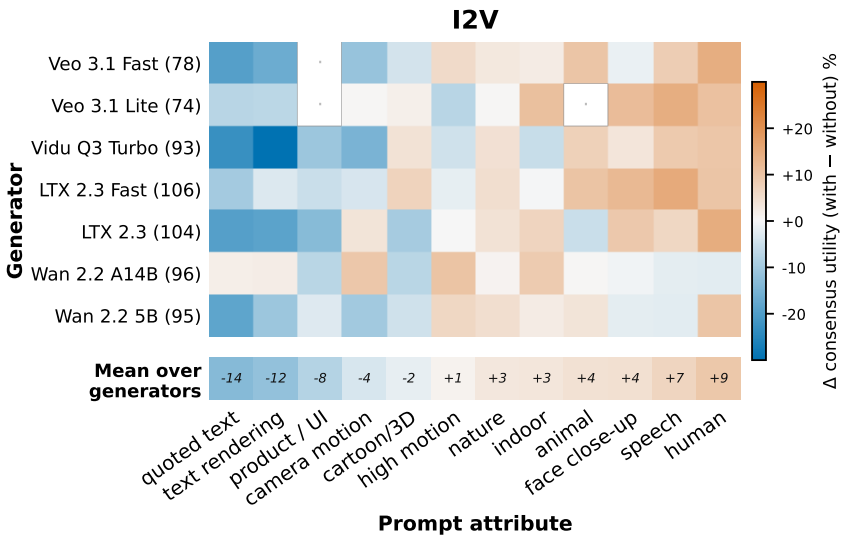



Figure 9: **Generator strength depends on the prompt (I2V).** The I2V half of Figure 1, on the same twelve attributes and with the same estimator, with numbers only in the mean row. The winner changes on 8 of the 12 attributes, mostly between the two Veo variants; on-screen text moves every generator at once ($-12$ to $-14$ on average, $-31$ for Vidu Q3 Turbo).

Table 2: Distribution of pool-level winning generators by consensus utility. Each of the 393 evaluable pools contributes one vote, with ties divided fractionally among co-winners. The five most frequent winners are shown; nine of the 30 generators each win more than 5% of the pools and 28 win at least one.

| Generator | Fractional pool wins | Win rate |
|---|---|---|
| ltx-2.3 | 39.7 | 10.10% |
| veo-3.1-fast | 34.8 | 8.84% |
| veo-3.1-lite-generate-001 | 31.2 | 7.95% |
| seedance-2.0-mini | 28.8 | 7.32% |
| veo-3.1-lite | 28.2 | 7.18% |

## C Additional Routing Results

This appendix carries the unabridged version of the routing figure and the two metrics the main text leaves out. Colors and markers follow the main text, and every panel uses the same full-candidate ground truth.

**All methods on NDCG.** Figure 11 repeats Figure 3 with every series: the three verifier variants of each agentic family, the held-out-label oracle, and the random baseline. The variants within a family differ by less than a point of NDCG, which is why the main text shows one of each. Octo-Human-Label reads the held-out labels, so it is a diagnostic ceiling rather than a method; on NDCG it sits at the bottom because a cascade never orders the candidates it did not generate.

**Top-1 and Top-3.** Top-1 and Top-3 over the same runs are below; the panels, budgets and baselines are identical, and every conclusion in Section 7 holds on all three metrics. Figure 12 gives Top-1 and Figure 13 gives Top-3. Top-1 is the noisiest of the three, since a single held-out pool contributes one binary outcome; Top-3 is the most saturated, with every learned method above 80% at most budgets, which is why the main text reports NDCG.

**Recovering opposing preferences.** Figure 14 is the evidence behind the opposite-preference test of Section 8.2. Each held-out pair is one on which two annotators expressed opposite preferences,

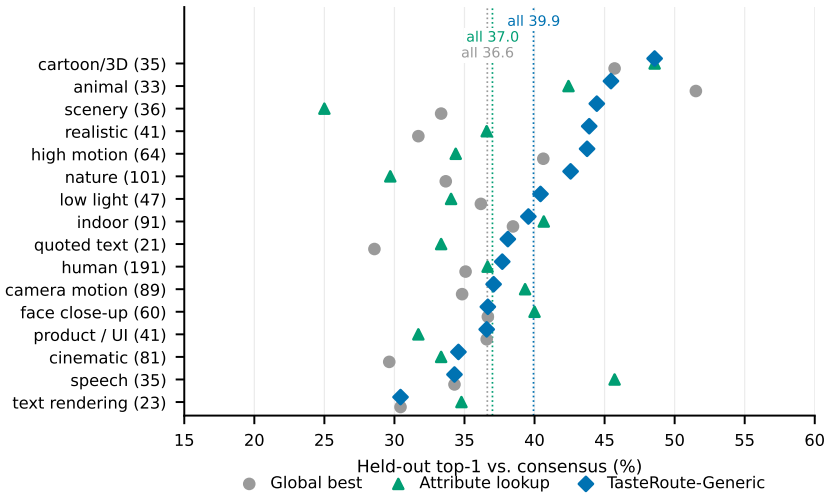



Figure 10: **TasteRoute-Generic by prompt attribute.** Held-out outer-fold T2V pools grouped by the keyword-detected attributes of Section 4, counts in parentheses, all graded against the pool's consensus ranking. The router is compared with two prompt-independent references: the single best generator on the training folds (*global best*) and the best generator for each attribute on the training folds (*attribute lookup*). Markers are offset vertically within a row and each dotted line is that series' rate over all attributes. Note that Figure 5 uses a different ground truth (each annotator's own ranking) and a different unit (annotator–pool pairs), so the two are not comparable.

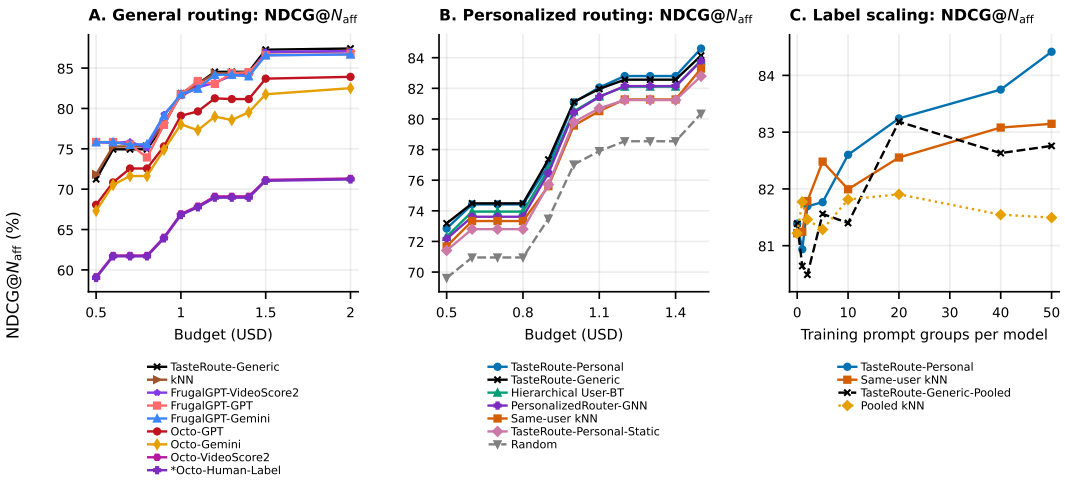



Figure 11: **NDCG across the three routing experiments, all methods**, laid out as Figure 3.

so prompt and candidates are identical for both and any router without annotator information scores 50% by construction. The floor is therefore a property of the design, not an estimate.

## D Cost-Supervision Trade-off

Section 7 states that explicit cost supervision buys savings only by giving up preference accuracy. Table 3 is the sweep behind that claim: a cost penalty $\lambda$ is added to the personalized training objective and the resulting router is evaluated at four budget caps. Reading down a column shows the price of each increment of saving. Penalties small enough to leave Top-1 untouched save almost nothing, and every configuration that saves more than a percent also loses accuracy, monotonically in $\lambda$. These runs are exploratory: they use the fixed T2V/I2V subset of the personalization test splits and a single seed triple, so they indicate the shape of the trade-off rather than a tuned operating point.

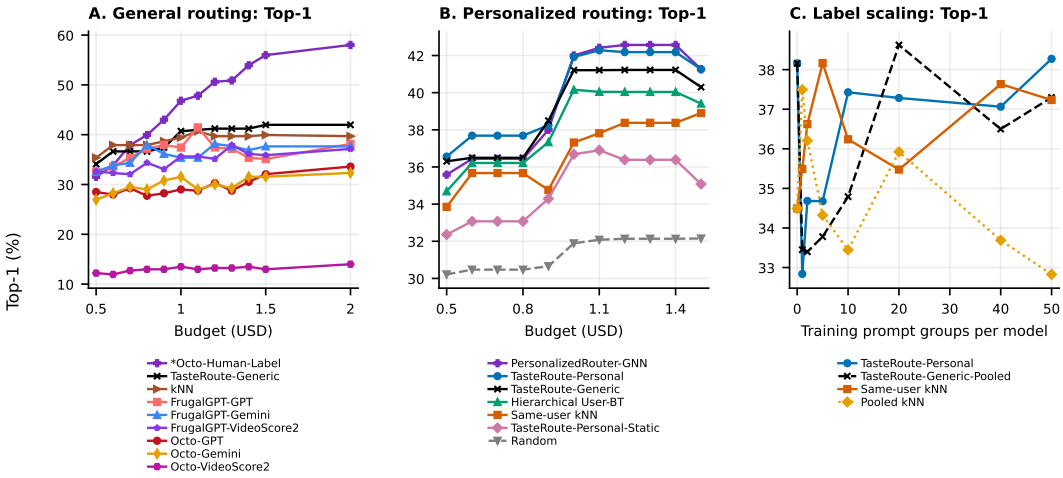



Figure 12: **Top-1 across the three routing experiments**, laid out as Figure 3. (A) General routing by budget. (B) Personalized routing by budget. (C) Matched-budget label scaling.

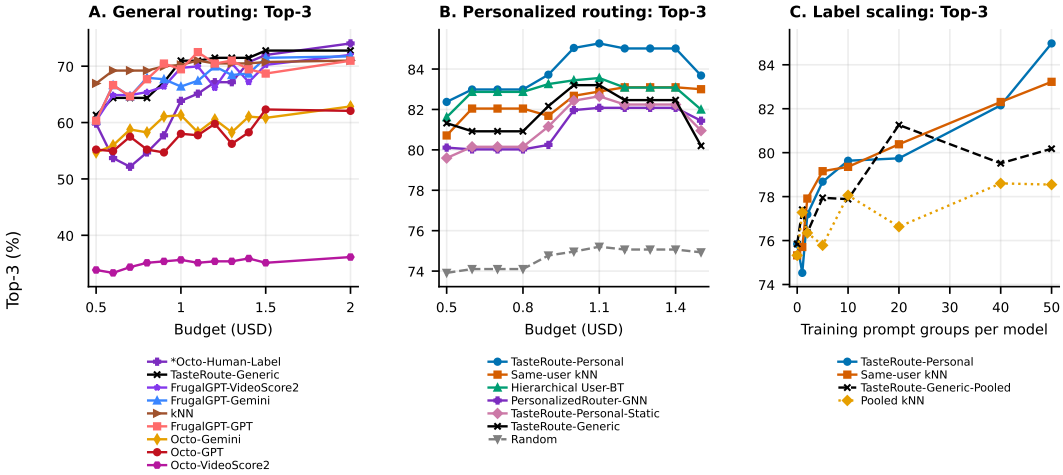



Figure 13: **Top-3 across the three routing experiments**, laid out as Figure 3. (A) General routing by budget. (B) Personalized routing by budget. (C) Matched-budget label scaling.

# E Related Works

## E.1 Model Routing

The idea of model-level routing, which selects a model for inference based on the input, dates back to classification settings (Kilic and Alpaydin, 2011; Aljundi et al., 2017), where routing was primarily used to maximize predictive performance. With the emergence of generative models such as LLMs and diffusion models, inference cost has become an additional consideration because serving increasingly capable models requires substantial computation. To maximize quality under a limited cost budget, Chen et al. (2024a); Ding et al. (2024) study cascade-based routing, while Hu et al. (2024) formalizes standardized quality–cost evaluation for model routers. Subsequent work generalizes routing through richer representations and decision mechanisms, including query-based representations (Chen et al., 2024b; Li, 2026), graph-based query–model interaction (Feng et al., 2025), multi-armed bandits (Rezaei et al., 2025), sequential multi-round routing (Zhang et al., 2025), and training-free routing (Tang et al., 2026). Beyond optimizing quality and cost, model routing can also support personalization by selecting different models for different users (Dai et al., 2025; Piskala et al., 2024).

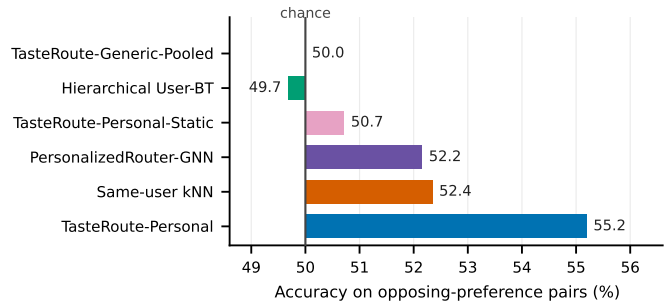



Figure 14: **Recovering opposing preferences.** Accuracy on held-out video pairs where two annotators disagree. A router without user information scores 50% by construction.

Table 3: Cost-preference trade-off for the personalized TasteRoute. Top-1 accuracy(%) and cost savings(%) are reported, averaged over nine annotators and three split seeds on a fixed T2V/I2V subset of the personalization test splits. Savings are relative to unchanged training ($\lambda = 0$) at the same budget cap. The utility-score formula is unchanged; $\lambda$ weights only the additional training cost loss.

| | \$0.50 cap | | \$1.00 cap | | \$1.50 cap | | \$1.70 cap | |
|---|---|---|---|---|---|---|---|---|
| $\lambda$ | Top-1% ↑ | Saving %↑ | Top-1% ↑ | Saving %↑ | Top-1% ↑ | Saving %↑ | Top-1% ↑ | Saving %↑ |
| 0 | 47.01 | 0.00 | 42.51 | 0.00 | 39.85 | 0.00 | 39.85 | 0.00 |
| 0.003 | 47.01 | 0.00 | 42.51 | 0.03 | 39.85 | 0.03 | 39.85 | 0.03 |
| 0.01 | 47.01 | 0.00 | 42.51 | 0.03 | 39.85 | 0.03 | 39.85 | 0.03 |
| 0.03 | 46.92 | 0.13 | 42.33 | 0.84 | 39.69 | 0.73 | 39.69 | 0.73 |
| 0.1 | 46.25 | 0.91 | 40.95 | 2.86 | 38.79 | 3.82 | 38.79 | 3.81 |
| 0.3 | 46.09 | 2.31 | 40.26 | 7.33 | 38.44 | 9.77 | 38.44 | 9.73 |
| 0.5 | 45.48 | 4.18 | 39.27 | 9.52 | 37.34 | 16.21 | 37.34 | 16.57 |
| 1.0 | 45.61 | 9.34 | 38.92 | 16.66 | 37.05 | 25.18 | 37.05 | 25.50 |

### E.2 Model Routing In The Age of Generative Model

Most of the field in model routing focuses on language models, some early work has begun to formulate model selection directly for image generation. PAK-UCB (Hu et al., 2025) casts prompt-aware generator selection as an online contextual-bandit problem under sequential feedback, using automatic fidelity metrics such as CLIPScore as rewards. More recently, OctoT2I (Jiang et al., 2026b) moves beyond one-shot selection with an agentic, self-evolving framework that autonomously discovers capability profiles across T2I models and uses verifier feedback to guide subsequent routing decisions.

Compared with language and image generation, model-level routing for video generation remains unexplored. This setting is particularly suited to routing because video generators differ substantially in both capability and serving cost, with reported 1080p image-to-video prices spanning roughly \$0.30–\$24.00 per generated minute (Artificial Analysis, 2026), while different users may also prefer different generators for the same request. These disparities motivate routing policies that select video generators according to the request and user rather than relying on a single backend.

## F Showcase of Labeling UI

We show the web interface used to collect TasteRoute-3K (Section 3). Each annotator first completes a six-step preference onboarding that initializes their preference profile (Figures 15–18), and then labels prompts in a ranking interface that collects both per-video quality ratings and listwise preference rankings (Figure 19).

**Preference onboarding.** The onboarding opens with a five-item questionnaire on self-reported preferences: the qualities that matter most (up to two), the preferred video type, motion amount,

camera treatment, and overall visual treatment (Figure 15). Throughout onboarding, a banner reminds annotators that there is no correct answer and that they should choose the result they would rather receive. To make this framing explicit, annotators then answer a quick comprehension check asking what the preference examples ask them to choose (Figure 16).

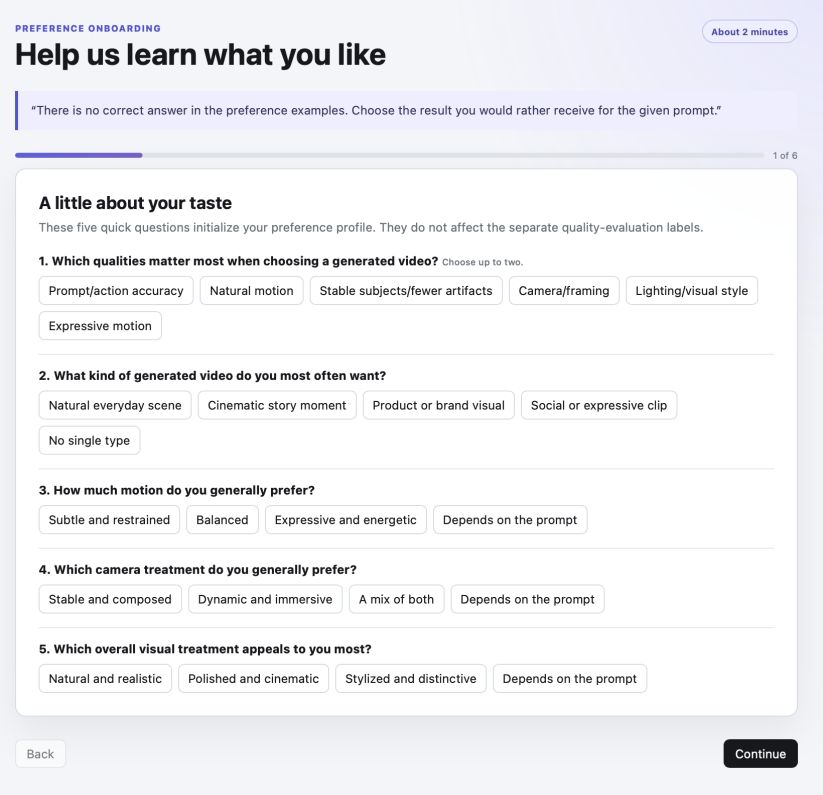



Figure 15: Onboarding step 1: the five-item preference questionnaire. Responses initialize the annotator's preference profile and do not affect the separate quality-evaluation labels.

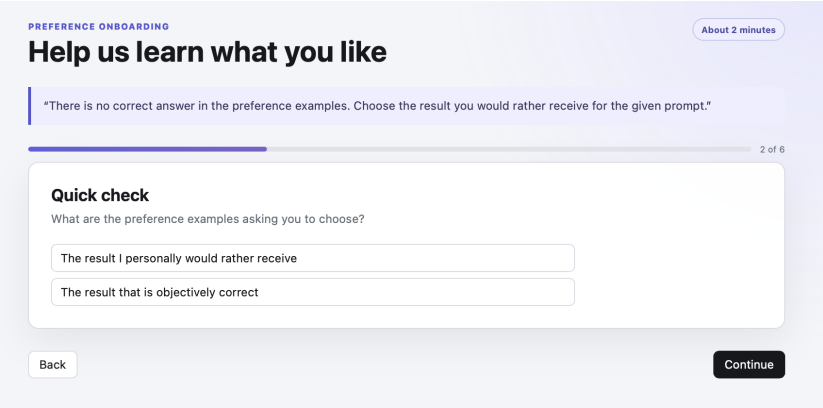



Figure 16: Onboarding step 2: comprehension check confirming that annotators understand the preference examples ask for the result they personally would rather receive, not an objectively correct one.

**Reference-video comparisons.** Because self-reported preferences may not reflect actual choices, the remaining onboarding steps present four reference-video comparisons. Annotators watch two generations for the same prompt and choose A, B, no preference, or cannot judge (Figure 17). The

follow-up questions depend on the choice (Figure 18): selecting A or B asks for up to two factors that influenced the choice and the preference strength (slight, moderate, strong); selecting no preference asks only for the influencing factors; and selecting cannot judge asks for a reason (playback problem, both videos unacceptable, difference not meaningful, or unclear prompt).

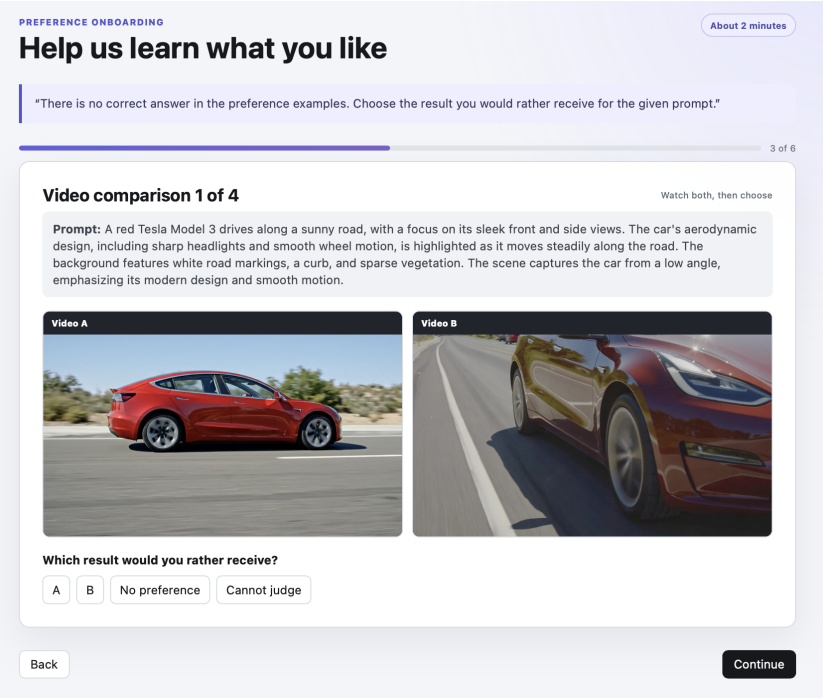
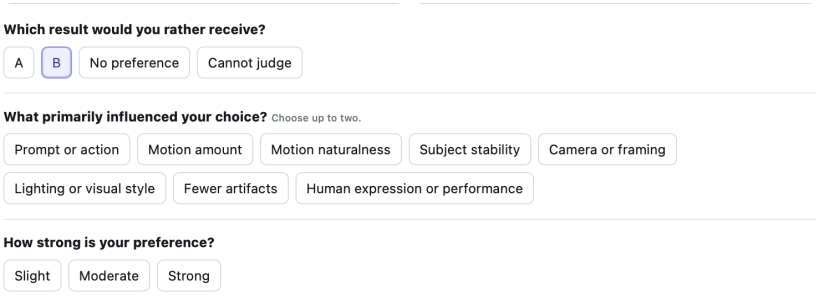



Figure 17: Onboarding steps 3–6: a reference-video comparison. Annotators watch both videos and choose which result they would rather receive.

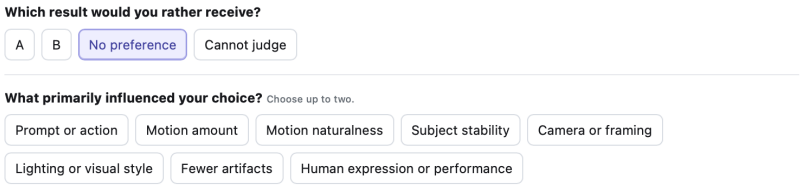
**Ranking interface.** After onboarding, annotators label prompts in the ranking interface (Figure 19). The left panel lists prompts by task (general or business, T2V or I2V) with their completion status, and the top of the main panel shows the conditioning prompt with its key entities highlighted. For each video, annotators first provide the five quality fields (text issues, motion artifacts, human realism, instruction following, and overall acceptability) on a 3-point scale. They then arrange the candidate videos into rank groups by dragging or using the up/down controls; placing several videos in one group expresses a tie, and a group can be split to break it.

## G Video Generators and Cost

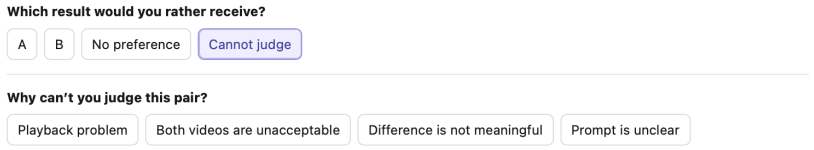
Table 4 lists the 30 video generators in the TasteRoute-3K candidate pool (Section 3), ordered by price. Prices are the catalog rates at the time of collection, in USD per second of 720p video without audio, and are frozen for all experiments (Appendix A). Some models are listed under two endpoint IDs, and each ID is treated as a separate candidate.




(a) Directional choice (A or B): influencing factors and preference strength.




(b) No preference: influencing factors only.




(c) Cannot judge: reason for abstaining.

Figure 18: Follow-up questions shown after each choice in a reference-video comparison.

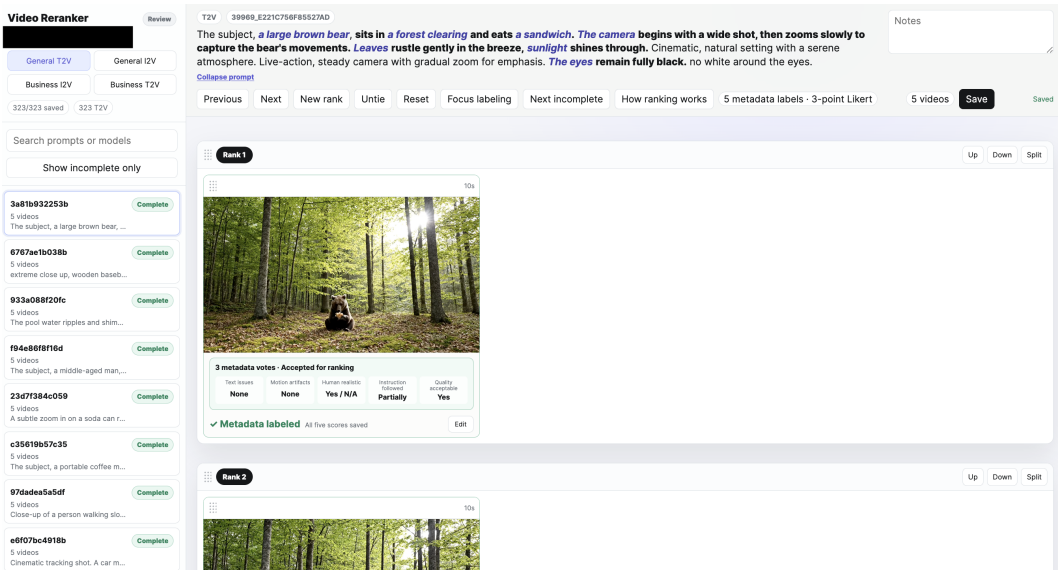



Figure 19: The ranking interface. Each video card shows its quality ratings; videos are ordered into rank groups, with ties allowed. Annotator-identifying information is redacted.

Table 4: Video generators in the candidate pool. *Tasks* gives the settings in which each generator appears in TasteRoute-3K. Prices are in USD per second of 720p video without audio; Veo and `gemini-omni-flash` prices follow Google's API catalogs, and all others follow fal.ai. †No 720p tier; billed at the 1080p rate. ‡Flat $0.15 per video, converted at 6 s. §Base price covers 5 s plus a per-second rate; shown as the 6 s equivalent.

| Model | Endpoint ID | Tasks | $/s |
|---|---|---|---|
| LongCat Video Distilled | `longcat-video-distilled-720p` | T2V | 0.010 |
| Wan 2.2 5B | `wan-v2.2-5b` | T2V, I2V | 0.025‡ |
| Veo 3.1 Lite | `veo-3.1-lite` | T2V, I2V | 0.030 |
| Veo 3.1 Lite | `veo-3.1-lite-generate-001` | T2V | 0.030 |
| LTX Video 13B Distilled | `ltx-video-13b-distilled` | T2V, I2V | 0.040 |
| Pika 2.2 | `pika-v2.2` | T2V, I2V | 0.040 |
| Kling 2.5 Turbo Standard | `kling-v2.5-turbo-standard` | I2V | 0.042§ |
| PixVerse V6 | `pixverse-v6` | T2V | 0.045 |
| PixVerse C1 | `pixverse-c1` | I2V | 0.050 |
| LTX 2.3 Fast | `ltx-2.3-fast` | T2V, I2V | 0.060† |
| Luma Ray 3.2 | `luma-ray-v3.2` | T2V, I2V | 0.060 |
| Grok Imagine Video | `grok-imagine-video` | T2V | 0.070 |
| Kling 2.5 Turbo Pro | `kling-v2.5-turbo-pro` | T2V | 0.070§ |
| Vidu Q3 Turbo | `vidu-q3-image-to-video-turbo` | I2V | 0.077 |
| Vidu Q3 Turbo | `vidu-q3-text-to-video-turbo` | T2V | 0.077 |
| LTX 2.3 Pro | `ltx-2.3` | T2V, I2V | 0.080† |
| Veo 3.1 Fast | `veo-3.1-fast` | T2V, I2V | 0.080 |
| Veo 3.1 Fast | `veo-3.1-fast-generate-001` | T2V | 0.080 |
| Wan 2.2 A14B | `wan-v2.2-a14b` | T2V, I2V | 0.080 |
| LTX 2.5 Fast | `ltx-2.5-fast` | T2V | 0.090 |
| Gemini Omni Flash | `gemini-omni-flash` | T2V, I2V | 0.100 |
| Gemini Omni Flash 1.1 | `gemini-omni` | T2V | 0.100 |
| Grok Imagine Video 1.5 | `grok-imagine` | T2V | 0.140 |
| Grok Imagine Video 1.5 | `grok-imagine-video-v1.5` | I2V | 0.140 |
| Happy Horse 1.1 | `happy-horse-v1.1` | T2V, I2V | 0.140 |
| Seedance 2.0 Mini | `seedance-2.0-mini` | T2V, I2V | 0.1547 |
| FLUX 3 | `flux-3` | T2V | 0.170 |
| Veo 3.1 | `veo-3.1` | T2V | 0.200 |
| Veo 3.1 | `veo-3.1-generate-001` | T2V, I2V | 0.200 |
| Seedance 2.0 Fast | `seedance-2.0-fast` | T2V, I2V | 0.2419 |